

CODIFICACIÓN DE COEFICIENTES PARA EL MODO DE SALTO DE TRANSFORMACIÓN

[0001] La presente solicitud reivindica el beneficio de:
solicitud de Patente US 16/814.654, presentada el 10 de marzo de 2020;
solicitud de patente provisional US 62/816.745, presentada el 11 de marzo de 2019; y
solicitud de patente provisional US 62/850.453, presentada el 20 de mayo de 2019,
cuyo contenido completo de cada una se incorpora 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, que incluyen televisores digitales, sistemas de transmisión digital directa, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio satelital, los llamados “teléfonos inteligentes”, dispositivos de videoconferencia, dispositivos de transmisión de video en 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

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

SÍNTESIS

[0005] En algunos escenarios de codificación, un codificador de video puede codificar datos de video en un modo de salto de transformación en el que no se realiza el proceso de transformación, es decir, se salta el proceso de transformación. Por tanto, para un bloque codificado en un modo de salto de transformación, los datos residuales no se transforman. La presente divulgación describe técnicas para un esquema de codificación de coeficientes para el modo de salto de transformación. Las técnicas de la presente divulgación incluyen un proceso de decodificación por entropía que convierte una representación binaria de coeficientes en una serie de coeficientes cuantificados con valor no binario. El correspondiente proceso de codificación por entropía, que generalmente es el proceso inverso de la decodificación por entropía, también forma parte de la presente divulgación.

[0006] En un ejemplo, un método de decodificación de datos de video incluye, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, determinar un valor para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado.

[0007] En otro ejemplo, un método de codificación de datos de video incluye, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, determinar un valor para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente

vecino; y codificar un valor para el coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado.

[0008] En otro ejemplo, un dispositivo para decodificar datos de video incluye una memoria configurada para almacenar datos de video y uno o más procesadores implementados en circuitos y configurados para: determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado.

[0009] En otro ejemplo, un dispositivo para codificar datos de video incluye una memoria configurada para almacenar datos de video y uno o más procesadores implementados en circuitos y configurados para: determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y codificar un valor para el coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado.

[0010] En otro ejemplo, un aparato para decodificar datos de video incluye medios para determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; medios para determinar un valor para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; medios para determinar un desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y medios para decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado.

[0011] En otro ejemplo, un aparato para codificar datos de video incluye medios para determinar, para un bloque residual de datos de video codificados que usa un modo de

salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; medios para determinar un valor para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; medios para determinar un desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y medios para codificar un valor para el coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado.

[0012] En otro ejemplo, un medio de almacenamiento legible por computadora almacena instrucciones que cuando son ejecutadas por uno o más procesadores hacen que uno o más procesadores determinen, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado.

[0013] En otro ejemplo, un medio de almacenamiento legible por computadora almacena instrucciones que cuando son ejecutadas por uno o más procesadores hacen que uno o más procesadores determinen, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y codificar un valor para el coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado.

[0014] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas serán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0015] 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.

[0016] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario cuaternario (QTBT) y una unidad de árbol de codificación (CTU) correspondiente.

[0017] La FIG. 3 muestra un ejemplo de coeficientes vecinos de un coeficiente que se está codificando o decodificando actualmente.

[0018] La FIG. 4 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[0019] La FIG. 5 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[0020] Las FIG. 6A y 6B son diagramas conceptuales que ilustran un proceso de actualización de intervalo en codificación aritmética binaria.

[0021] La FIG. 7 es un diagrama conceptual que ilustra un proceso de salida en codificación aritmética binaria.

[0022] La FIG. 8 es un diagrama de bloques que ilustra un codificador de codificación aritmética binaria adaptativa al contexto (CABAC) en un codificador de video.

[0023] La FIG. 9 es un diagrama de bloques que ilustra un codificador CABAC en un decodificador de video.

[0024] La FIG. 10 es un diagrama de flujo que ilustra un proceso para codificar datos de video.

[0025] La FIG. 11 es un diagrama de flujo que ilustra un proceso para decodificar datos de video.

[0026] La FIG. 12 es un diagrama de flujo que ilustra un proceso para determinar un contexto para codificar un signo de un coeficiente de un bloque residual.

DESCRIPCIÓN DETALLADA

[0027] La codificación de video (por ejemplo, codificación de video y/o decodificación de video) típicamente implica predecir un bloque de datos de video desde un bloque de datos de video ya codificado en la misma imagen (por ejemplo, intrapredicción) o un bloque de datos de video ya codificado en una imagen diferente (por ejemplo, interpredicción). En algunos casos, el codificador de video también calcula los datos residuales al comparar el bloque predictivo con el bloque original. Por tanto, los datos residuales representan una diferencia entre el bloque predictivo y el bloque original. Para reducir el número de bits necesarios para señalar los datos residuales, el codificador de video puede transformar y cuantificar los datos residuales y señalar los datos residuales transformados y cuantificados en el flujo de bits codificado. La compresión lograda por los procesos de transformación y cuantificación puede tener pérdidas, lo

que significa que los procesos de transformación y cuantificación pueden introducir distorsión en los datos de video decodificados

[0028] Un decodificador de video decodifica y agrega los datos residuales al bloque de predicción para producir un bloque de video reconstruido que coincida con el bloque de video original más de cerca que el bloque predictivo solo. Debido a la pérdida introducida por la transformación y cuantificación de los datos residuales, el bloque reconstruido puede tener distorsiones o interferencias. Un tipo común de interferencia o distorsión se conoce como bloqueo, donde los límites de los bloques utilizados para codificar los datos de video son visibles.

[0029] Para mejorar aún más la calidad del video decodificado, un decodificador de video puede realizar una o más operaciones de filtrado en los bloques de video reconstruidos. Los ejemplos de estas operaciones de filtrado incluyen el filtrado de desbloqueo, el filtrado de desplazamiento adaptativo de muestra (SAO) y el filtrado de bucle adaptativo (ALF). Los parámetros para estas operaciones de filtrado pueden determinarse mediante un codificador de video y señalizarse explícitamente en el flujo de bits de video codificado o pueden determinarse implícitamente mediante un decodificador de video sin necesidad de que los parámetros se señalicen explícitamente en el flujo de bits de video codificado.

[0030] En algunos escenarios de codificación, un codificador de video puede codificar datos de video en un modo de salto de transformación en el que no se realiza el proceso de transformación descrito con anterioridad, es decir, se salta el proceso de transformación. Por tanto, para un bloque codificado en un modo de salto de transformación, los datos residuales no se transforman. Un bloque residual de datos de video codificados que usa un modo de salto de transformación también puede denominarse bloque residual no transformado. La presente divulgación describe técnicas para un esquema de codificación de coeficientes para el modo de salto de transformación. Las técnicas de la presente divulgación incluyen un proceso de decodificación por entropía que convierte una representación binaria en una serie de coeficientes cuantificados con valores no binarios. El correspondiente proceso de codificación por entropía, que generalmente es el proceso inverso de la decodificación por entropía, también forma parte de la presente divulgación. Las técnicas de la presente divulgación se pueden aplicar a cualquiera de los códecs de video existentes, como Codificación de video de alta eficiencia (HEVC), o a un estándar que se está desarrollando actualmente, como Codificación de video versátil (VVC), y a otros estándares futuros de codificación de video.

[0031] La presente divulgación propone técnicas que incluyen, por ejemplo, determinar un desplazamiento de contexto para un coeficiente que se está decodificando actualmente en función de un valor para un primer coeficiente vecino y un valor para un segundo coeficiente vecino y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado. Como los valores de los coeficientes entre los coeficientes vecinos en los bloques residuales tienden a estar más correlacionados para los bloques de salto de transformación que para los bloques transformados, las técnicas de la presente divulgación pueden dar como resultado una codificación por entropía mejorada, lo que puede mejorar la eficiencia de codificación general, por ejemplo, al reducir la sobrecarga de bits necesaria para representar datos de video codificados sin degradar la calidad de los datos de video decodificados.

[0032] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video 100 que puede realizar las técnicas de la presente divulgación. Las técnicas de la presente divulgación están generalmente dirigidas a la codificación (codificación y/o decodificación) de datos de video. En general, los datos de video incluyen cualquier dato para procesar un video. Por lo tanto, los datos de video pueden incluir video sin codificar, video codificado, video decodificado (por ejemplo, reconstruido) y metadatos de video, como datos de señalización.

[0033] 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 ser cualquiera de una amplia gama de dispositivos o incorporarlos, incluyendo una computadora de escritorio, una *notebook* (es decir, una computadora portátil), tableta u otro dispositivo móvil, decodificador, auricular de teléfono, teléfono inteligente, televisión, cámara, dispositivo de visualización, reproductor de medios digitales, consola de videojuegos, dispositivo de transmisión de video en continuo, dispositivo receptor de transmisión o similar. En algunos casos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden estar equipados para comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[0034] 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 la presente divulgación, el codificador de video 200 del dispositivo de origen 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para codificación de coeficientes descrita en la presente divulgación. 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.

[0035] 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 codificación de coeficientes descritas en la presente divulgación. 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. La presente 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 modo que cada uno de los dispositivos 102 y 116 incluya 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 en continuo, reproducción de video, transmisión de video o telefonía de video.

[0036] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin procesar y 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.

[0037] La memoria 106 del dispositivo de origen 102 y la memoria 120 del dispositivo de destino 116 representan memorias de uso general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video sin procesar de la fuente de video 104 y datos de video sin procesar y sin decodificar 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 el decodificador de video 300, respectivamente. Aunque la memoria 106 y la memoria 120 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 una o más memorias intermedias de video, por ejemplo, para almacenar datos de video sin procesar, decodificados y/o codificados.

[0038] 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 a 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, de acuerdo con un estándar de comunicación inalámbrica, puede modular una señal de transmisión que incluya los datos de video

codificados, y la interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 102 al dispositivo de destino 116.

[0039] En algunos ejemplos, el medio legible por computadora 110 puede incluir el dispositivo de almacenamiento 112. El dispositivo de origen 102 puede emitir datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a los datos codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados.

[0040] En algunos ejemplos, el medio legible por computadora 110 puede incluir el servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que pueda almacenar los datos de video codificados generados por el dispositivo de origen 102. 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 datos de video codificado generados 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 en 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 protocolo de transferencia de archivos (FTP), un dispositivo de red de entrega de contenido o un dispositivo de almacenamiento conectado a la red (NAS). El dispositivo de destino 116 puede acceder a datos de video codificados desde el servidor de archivos 114 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión

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

[0041] 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 (Evolución a largo plazo), LTE avanzada, 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.

[0042] Las técnicas de la presente divulgación se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de transmisión en continuo por Internet, tales como transmisión en continuo 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.

[0043] 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, un medio de comunicación, dispositivo de almacenamiento 112, servidor de archivos 114 o similar).

El flujo de bits de video codificado puede incluir información de signo definida por el codificador de video 200, que también es utilizado por el decodificador de video 300, como elementos de sintaxis que tienen valores que describen características y/o el procesamiento de bloques de video u otras unidades codificadas (por ejemplo, segmentos, imágenes, grupos de imágenes, secuencias o similares). El dispositivo de visualización 118 muestra imágenes decodificadas de los datos de video decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánica (OLED) u otro tipo de dispositivo de visualización.

[0044] Aunque no se muestra en la FIG. 1, en algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar integrados con un codificador de audio y/o decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro *hardware* y/o *software*, para manejar flujos multiplexados que incluyen ambos, audio y video, en un flujo de datos común. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (UDP).

[0045] El codificador de video 200 y el decodificador de video 300 pueden implementarse como cualquiera de una variedad de circuitos de codificador y/o decodificador adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados específicos de aplicación (ASIC), matrices de puerta programables en campo (FPGA), lógica discreta, *software*, *hardware*, *firmware* o cualquier combinación de 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* mediante el uso de uno o más procesadores para realizar las técnicas de la presente divulgación. Cada codificador de video 200 y decodificador de video 300 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de video 200 y/o un decodificador de video 300 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

[0046] El codificador de video 200 y el decodificador de video 300 pueden funcionar 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

la vista múltiple y/o extensiones de codificación de video escalables. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden funcionar 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 del estándar de 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-v5 (en adelante, “Borrador 4 de VVC”). Otro borrador del estándar de VVC se describe en Bross, *et al.* “Versatile Video Coding (Draft 7)”, Equipo conjunto de expertos en video (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 16ª reunión: Ginebra, CH, 1-11 de octubre de 2019, JVET-P2001-v14 (en adelante, “borrador 7 de VVC”). Las técnicas de la presente divulgación, sin embargo, no se limitan a ningún estándar de codificación particular.

[0047] 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.

[0048] La presente 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, la presente 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 partición 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.

[0049] La HEVC define varios bloques, que incluyen unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). De acuerdo con HEVC, un codificador de video (como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU según una estructura de árbol cuaternario. Es decir, el codificador de video divide las CTU y CU en cuatro cuadrados iguales y no superpuestos, y cada nodo del árbol cuaternario 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 dividir más PU y TU. Por ejemplo, en HEVC, un árbol cuaternario residual (RQT) representa la partición de TU. En HEVC, las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU intrapredichas incluyen información de intrapredicción, como una indicación de intramodo.

[0050] 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 cuaternario (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 dividido de acuerdo con la partición de árbol cuaternario y un segundo nivel dividido de acuerdo con la partición de árbol binario. Un nodo raíz de la estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a unidades de codificación (CU).

[0051] En una estructura de partición MTT, los bloques pueden dividirse mediante una partición de árbol cuaternario (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT) (también llamado árbol ternario (TT)). Una partición de árbol triple o ternario es una partición en la que un bloque se divide en tres subbloques. En algunos ejemplos, una partición de árbol triple o ternario divide un bloque en tres subbloques sin dividir el bloque original a través del centro. Los tipos de

partición en MTT (por ejemplo, QT, BT y TT) pueden ser simétricos o asimétricos.

[0052] 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).

[0053] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar la partición de árbol cuaternario por HEVC, partición QTBT, partición MTT u otras estructuras de partición. 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 de árbol cuaternario u otros tipos de partición.

[0054] Los bloques (por ejemplo, CTU o CU) pueden agruparse de varias formas en una imagen. Como un ejemplo, un ladrillo puede referirse a una región rectangular de filas de CTU dentro de un mosaico particular en una imagen. Un mosaico puede ser una región rectangular de CTU dentro de una columna de mosaico particular y una fila de mosaico particular en una imagen. Una columna de mosaico se refiere a una región rectangular de CTU que tiene una altura igual a la altura de la imagen y un ancho especificado por elementos de sintaxis (por ejemplo, como en un conjunto de parámetros de imagen). Una fila de mosaico se refiere a una región rectangular de CTU que tiene una altura especificada por elementos de sintaxis (por ejemplo, en un conjunto de parámetros de imagen) y un ancho igual al ancho de la imagen.

[0055] En algunos ejemplos, un mosaico puede dividirse en múltiples ladrillos, cada uno de los cuales puede incluir una o más filas de CTU dentro del mosaico. Un mosaico que no está dividido en múltiples ladrillos también puede denominarse ladrillo. Sin embargo, un ladrillo que es un verdadero subconjunto de un mosaico no puede denominarse mosaico.

[0056] Los ladrillos en una imagen también pueden estar dispuestos en un segmento. Un segmento puede ser un número entero de ladrillos de una imagen que puede estar contenido exclusivamente en una sola unidad de capa de abstracción de red (NAL). En algunos ejemplos, un corte incluye una cantidad de mosaicos completos o solo una secuencia consecutiva de ladrillos completos de un mosaico.

[0057] La presente 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 un CU pueden organizarse en filas y columnas. Además, las CU no necesitan necesariamente tener el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras de NxM, donde M no es necesariamente igual a N.

[0058] 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 la CU antes de la codificación y el bloque de predicción.

[0059] 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 mediante 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 mediante 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 mediante el uso de predicción unidireccional o predicción bidireccional.

[0060] Algunos ejemplos de JEM y VVC también proporcionan un modo de compensación de movimiento afín, que puede considerarse un modo de interpredicción. En el modo de compensación de movimiento afín, el codificador de video 200 puede

determinar dos o más vectores de movimiento que representan movimiento no traslacional, tal como acercamiento o alejamiento, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0061] 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 escaneo de trama (de izquierda a derecha, de arriba a abajo).

[0062] 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 mediante 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 afín.

[0063] 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 mediante el uso del modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformaciones al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada discreta de coseno (DCT), una transformada entera, una transformada ondícula o una transformada conceptualmente similar a los datos de video residual. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformación, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada Karhunen-Loeve (KLT) o

similar. El codificador de video 200 produce coeficientes de transformación después de la aplicación de una o más transformaciones.

[0064] Aunque lo anterior describe ejemplos en los que se realizan transformaciones, en algunos ejemplos, la transformación puede omitirse. Por ejemplo, el codificador de video 200 puede implementar el modo de salto de transformación en el que se salta la operación de transformación. En los ejemplos en los que se omite la transformación, el codificador de video 200 puede generar coeficientes correspondientes a valores residuales en lugar de coeficientes de transformación. En la siguiente descripción, el término “coeficiente” debe interpretarse como que incluye coeficientes correspondientes a valores residuales o coeficientes de transformación generados a partir del resultado de una transformación.

[0065] Como se indicó con anterioridad, después de cualquier transformación o cuando se omite la transformación para producir coeficientes, el codificador de video 200 puede realizar la cuantificación de los coeficientes. En algunos ejemplos, la cuantificación también se puede omitir cuando se omite la transformación. La cuantificación generalmente se refiere a un proceso en el que los coeficientes se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una mayor compresión. 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 por cuantificar.

[0066] Después de la cuantificación, el codificador de video 200 puede escanear los coeficientes (por ejemplo, generados a partir del resultado de la transformación o debido al salto de transformación), produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes cuantificados. El escaneo puede diseñarse para colocar coeficientes de energía más alta (y por lo tanto de frecuencia más baja) al frente del vector y para colocar coeficientes de energía más baja (y por lo tanto de frecuencia más alta) en la parte posterior del vector. En los ejemplos en los que se omite la transformación, el resultado del escaneo puede no ser que los coeficientes de energía más altos estén al frente del vector y los coeficientes de energía más bajos estén en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede utilizar un orden de escaneo predefinido para escanear los coeficientes cuantificados para producir un vector serializado, y luego codificar por entropía los coeficientes

cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar un escaneo adaptativo. Después de escanear los coeficientes 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 su uso por parte del decodificador de video 300 para decodificar los datos de video.

[0067] Como se introdujo con anterioridad, el codificador de video 200 codifica datos residuales en TU. Según las características esperadas de los datos residuales en una TU, el codificador de video 200 puede codificar las TU en diferentes modos, tales como un modo de transformación o un modo de salto de transformación, con diferentes modos que utilizan diferentes esquemas de codificación de coeficientes. Algunos esquemas de codificación de coeficientes utilizan grupos de coeficientes para codificar una TU. Un grupo de coeficientes generalmente es un subconjunto de los coeficientes en una TU. Por ejemplo, el codificador de video 200 puede codificar una TU de 16x16 como cuatro grupos de coeficientes de 4x4.

[0068] Para realizar CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede relacionarse, por ejemplo, si los valores vecinos del símbolo tienen valor cero o no. La determinación de probabilidad puede basarse en un contexto asignado al símbolo.

[0069] 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.

[0070] De esta manera, el codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados, por ejemplo, elementos de sintaxis que describen la partición de una imagen en bloques (por ejemplo, CU) y predicción y/o información residual para los bloques. Finalmente, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[0071] 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 mediante 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.

[0072] La información residual puede representarse, por ejemplo, por coeficientes cuantificados que representan valores residuales o coeficientes de transformación. 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. En los ejemplos en los que el codificador de video 200 omitió la operación de transformación (por ejemplo, el modo de salto de transformación), el decodificador de video 300 puede omitir la operación de transformación inversa. 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 (es decir, bloque predictivo) 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 la interferencia visual a lo largo de los límites del bloque.

[0073] Según las técnicas de la presente divulgación, el codificador de video 200 y el decodificador de video 300 pueden configurarse para determinar un desplazamiento de contexto para un coeficiente que se está decodificando actualmente en función de un valor para un primer coeficiente vecino y un valor para un segundo coeficiente vecino y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado. Como los valores de los coeficientes entre los coeficientes vecinos en los bloques residuales tienden a estar más correlacionados para los bloques de salto de transformación que para los bloques transformados, las técnicas de la presente divulgación pueden dar como resultado una codificación por entropía mejorada, lo que puede mejorar la eficiencia de codificación general, por ejemplo, al reducir la sobrecarga de bits necesaria para representar datos

de video codificados sin degradar la calidad de los datos de video decodificados.

[0074] La presente divulgación generalmente puede referirse a “señalizar” cierta información, tal como elementos de sintaxis. El término “señalización” puede referirse generalmente a la comunicación de valores para elementos de sintaxis y/u otros datos 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 parte del dispositivo de destino 116.

[0075] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario cuaternario (QTBT) 130, y una unidad de árbol de codificación correspondiente (CTU) 132. Las líneas continuas representan la división de árbol cuaternario y las líneas punteadas 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 árbol cuaternario, no es necesario indicar el tipo de división, ya que los nodos de árbol cuaternario dividen un bloque horizontal y verticalmente en 4 subbloques 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 (es decir, el primer nivel) de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (tal como información de división) para un nivel de árbol de predicción (es decir, el segundo nivel) 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 predicción y transformación de datos, para CU representadas por nodos hoja terminales de la estructura QTBT 130.

[0076] En general, CTU 132 de la FIG. 2B puede estar asociado con parámetros que definen tamaños de bloques correspondientes a nodos de la estructura QTBT 130 en el primer y segundo nivel. Estos parámetros pueden incluir un tamaño de CTU (que representa un tamaño de CTU 132 en muestras), un tamaño mínimo de árbol cuaternario (MinQTSIZE, que representa un tamaño mínimo permitido de nodo hoja de árbol cuaternario), un tamaño máximo de árbol binario (MaxBTSIZE, que representa un

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

[0077] 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 árbol cuaternario. 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 nodos tales como el nodo primario y los nodos secundarios que tienen líneas continuas para las ramas. Si los nodos del primer nivel no son mayores que el tamaño máximo permitido del nodo raíz del árbol binario (MaxBTSize), entonces los nodos pueden dividirse aún más por los respectivos árboles binarios. La división del árbol binario de un nodo puede repetirse hasta que los nodos resultantes de la división alcancen el tamaño mínimo permitido del nodo hoja del árbol binario (MinBTSize) o la profundidad máxima permitida del árbol binario (MaxBTDepth). El ejemplo de la estructura QTBT 130 representa los nodos que tienen líneas discontinuas para las ramas. El nodo 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 se transforma, sin ninguna partición adicional. Como se explicó con anterioridad, las CU también pueden denominarse “bloques de video” o “bloques”.

[0078] En un ejemplo de la estructura de partición de QTBT, el tamaño de la CTU se establece en 128x128 (muestras de luma y dos muestras de croma 64x64 correspondientes), el MinQTSIZE se establece en 16x16, el MaxBTSize se establece en 64x64, el MinBTSize (para ambos anchos y altura) se establece en 4 y el MaxBTDepth se establece en 4. La partición de árbol cuaternario se aplica primero a la CTU para generar nodos hoja de árbol cuaternario. Los nodos hoja de árbol cuaternario 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 árbol cuaternario es 128x128, el nodo hoja de árbol cuaternario no será más dividido por el árbol binario, ya que el tamaño excede el MaxBTSize (es decir, 64x64, en este ejemplo). De lo contrario, el árbol binario dividirá el nodo hoja de árbol cuaternario. Por lo tanto, el nodo hoja de árbol cuaternario es también el nodo raíz del árbol binario y tiene la profundidad del árbol binario en 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.

[0079] Cuando un bloque de datos de video se codifica en un modo de salto de transformación, el codificador de video 200 omite el proceso de transformada para señales residuales antes de realizar un proceso de cuantificación. El decodificador de video 300 también omite el paso del proceso de transformada inversa después de realizar un proceso de descuantificación. Las características de una señal residual no transformada son típicamente bastante diferentes a las de las señales transformadas. Por ejemplo, los coeficientes de un bloque de salto de transformación tienden a estar más correlacionados con sus coeficientes vecinos en comparación con los coeficientes de un bloque transformado. Como resultado, los valores de nivel y la información de signo para coeficientes vecinos en un bloque de salto de transformación de datos residuales tienden a estar más correlacionados en comparación con los valores de nivel y la información de signo para un bloque transformado de datos residuales.

[0080] B. Bross, T. Nguyen, P. Keydel, H. Schwarz, D. Marpe, T. Wiegand, “Non-CE8: Unified Transform Type Signalling and Residual Coding for Transform Skip”, documento JVET JVET-M0464, Marrakech, MA, enero de 2019, establecen un proceso propuesto para realizar codificación residual para bloques codificados en un modo de salto de transformación. Para una codificación eficiente de los niveles y la información de signo en el modo de salto de transformación, la codificación de coeficiente propuesta en JVET-M0464 puede modificarse para aprovechar las características de la señal para una codificación más eficiente.

[0081] La FIG. 3 muestra un ejemplo de tres coeficientes de un bloque de salto de transformación de datos residuales. El bloque de salto de transformación también incluiría coeficientes adicionales que no se muestran en la FIG. 3. En el ejemplo de la FIG. 3, el valor del coeficiente X representa el valor del coeficiente 140, que representa un coeficiente que se está codificando actualmente. El valor del coeficiente X0 representa el valor del coeficiente 142, que es el coeficiente vecino izquierdo del coeficiente 140. El valor del coeficiente X1 representa el valor del coeficiente 144, que es el coeficiente vecino superior del coeficiente 140. En la presente divulgación, el vecino superior también puede denominarse vecino anterior.

[0082] El codificador de video 200 y el decodificador de video 300 pueden configurarse

para realizar la codificación de señalización para los coeficientes en un bloque de salto de transformación. En las técnicas para la codificación residual de salto de transformación descritas en JVET-M0464, los coeficientes se codifican desde el primero (es decir, arriba a la izquierda) al último (es decir, abajo a la derecha) en lugar del último al primero como se hace para los bloques transformados. Si se implementan las técnicas de JVET-M0464, el codificador de video 200 y el decodificador de video 300 pueden configurarse para codificar el contexto, mediante el uso de CABAC, la información de signo mediante el uso de un tipo de canal de residual como contexto. Es decir, el codificador de video 200 y el decodificador de video 300 pueden utilizar un contexto para luma residual y otro contexto para croma residual. La presente divulgación describe técnicas de codificación de señal que pueden usarse en lugar de o además de las técnicas descritas en JVET-M0464. Las siguientes técnicas utilizan la información de signo de un coeficiente vecino anterior (por ejemplo, el coeficiente 144 en la FIG. 3) y un coeficiente vecino izquierdo (por ejemplo, el coeficiente 142 en la FIG. 3) para derivar el desplazamiento del contexto de codificación de señal para un coeficiente que se está codificando actualmente (por ejemplo, coeficiente 140 en la FIG. 3).

[0083] En referencia a la FIG. 3, X0 es el valor del coeficiente vecino izquierdo y X1 es el valor del coeficiente vecino anterior. Si ambos coeficientes vecinos son cero o ambos distintos a cero, pero con signos opuestos, entonces el codificador de video 200 y el decodificador de video 300 pueden usar un desplazamiento de contexto 0 (ctxOffset = 0). De lo contrario, si ambos son no negativos, entonces el codificador de video 200 y el decodificador de video 300 pueden usar el desplazamiento de contexto 1 (ctxOffset = 1). Para todos los demás casos, el codificador de video 200 y el decodificador de video 300 pueden utilizar el desplazamiento de contexto 2 (ctxOffset = 2). Esto se puede describir mediante el siguiente pseudocódigo:

```

    if ((X0 == 0 && X1 == 0) || ((X0 * X1) < 0))
    {
        ctxOffset = 0;
    }
    else if (X0 >= 0 && X1 >= 0)
    {
        ctxOffset = 1;
    }
    else
    {

```

```

        ctxOffset = 2;
    }

```

[0084] En algunos ejemplos, si ambos coeficientes vecinos son cero o ambos distintos a cero, pero con signos opuestos, entonces el codificador de video 200 y el decodificador de video 300 pueden usar el desplazamiento de contexto 0. De lo contrario (cuando ambos son positivos, o ambos son negativos, o uno es cero, y el otro es distinto de cero), el codificador de video 200 y el decodificador de video 300 pueden usar el contexto 1. Si se usa el contexto 1, entonces el valor del signo señalado de 0 o 1, según la convención, significaría que el signo del coeficiente que se está codificando actualmente es el mismo que el signo de uno de los vecinos distintos de cero. Esta técnica también se puede extender a un escenario de codificación donde solo se puede usar un valor de coeficiente distinto de cero codificado previamente para la derivación del contexto, donde un valor de signo de 0 o 1 indica que el signo del coeficiente es el mismo que el signo del signo de coeficiente distinto de cero anterior codificado con contexto único.

[0085] Los conjuntos de contextos separados para los componentes de luminancia y crominancia se pueden usar en combinación con la derivación de desplazamiento de contexto descrita con anterioridad.

[0086] El codificador de video 200 y el decodificador de video 300 también pueden configurarse para realizar la asignación de nivel. En la codificación residual de salto de transformación de JVET-M0464, los niveles absolutos del coeficiente `absCoeffLevel` se codifican mediante el uso del valor `sig_coeff_flag`, `abs_level_gtX_flags`, `par_level_flag` y `abs_remainder` para formar el valor del coeficiente de transformación absoluta final, donde X puede ser 1, ..., 5 (o algún otro valor de corte C). Entonces, el valor `absCoeffLevel` puede ser construido por:

$$\text{absCoeffLevel} = 1 + \text{abs_level_gt1_flag} + \text{par_level_flag} + 2 * (\text{abs_level_gt2_flag} + \text{abs_level_gt3_flag} + \dots + \text{abs_level_gtC_flag}) + 2 * \text{abs_remainder}$$

[0087] En lugar de representar el `absCoeffLevel` directamente como en JVET-M0464, el codificador de video 200 puede configurarse para asignar `absCoeffLevel` a un nivel modificado. El decodificador de video 300 puede configurarse para realizar una asignación inversa.

[0088] De forma similar a las técnicas de derivación de desplazamiento de contexto de codificación de signos descritas con anterioridad, el codificador de video 200 y el decodificador de video 300 pueden utilizar la información `absCoeffLevel` de los coeficientes vecinos izquierdo y superior para codificar y decodificar valores de nivel de coeficiente absoluto. En este ejemplo, X_0 denota el nivel de coeficiente absoluto de un

coeficiente vecino izquierdo (por ejemplo, el coeficiente 142 en la FIG. 3) al coeficiente que se está codificando actualmente (por ejemplo, el coeficiente 140 en la FIG. 3), y X_1 denota el nivel de coeficiente absoluto de un coeficiente vecino anterior (por ejemplo, el coeficiente 144 en la FIG. 3) al coeficiente que se está codificando actualmente (por ejemplo, el coeficiente 140 en la FIG. 3). Para representar un coeficiente con un nivel de coeficiente absoluto $absCoeff$, se puede codificar un $absCoeffMod$ asignado.

[0089] El funcionamiento del codificador de video 200 para derivar un valor para $absCoeffMod$ se puede mostrar con el siguiente pseudocódigo:

```
pred = max(X0, X1);  
    if (absCoeff == pred  
    {  
        absCoeffMod = 1;  
    }  
    else  
    {  
        absCoeffMod = (absCoeff < pred) ? absCoeff + 1 : absCoeff;  
    }
```

[0090] En algunos ejemplos, si el valor absoluto de un coeficiente ($absCoeff$) a codificar es igual al predictor vecino máximo, $pred$, entonces el codificador de video 200 establece el nivel modificado $absCoeffMod$ en 1. De lo contrario, si $absCoeff$ es menor que el predictor, entonces el codificador de video 200 incrementa el valor a codificar en 1. De lo contrario, el codificador de video 200 no modifica el valor $absCoeff$.

[0091] El codificador de video 200 puede determinar, por ejemplo, un valor de nivel predicho para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino y, en respuesta a que un valor de nivel del coeficiente que se está codificando actualmente es igual al valor de nivel predicho, codificar un elemento de sintaxis con un valor igual a uno. En otros casos, el codificador de video 200 puede determinar un valor de nivel predicho para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino y, en respuesta a que un valor de nivel del coeficiente que se está codificando actualmente es menor que el valor de nivel predicho, codifica un elemento de sintaxis con un valor igual al valor de nivel del coeficiente que se está codificando actualmente. En otros casos, el codificador de video 200 puede determinar un valor de nivel predicho para el coeficiente que se está codificando actualmente en función del valor para el primer

coeficiente vecino y el valor para el segundo coeficiente vecino y, en respuesta a que un valor de nivel del coeficiente que se está codificando actualmente es mayor que el valor de nivel predicho, codificar un elemento de sintaxis con un valor igual al valor de nivel del coeficiente que se está codificando actualmente menos uno.

[0092] El funcionamiento del decodificador de video 300 para derivar un valor de absCoeff se puede mostrar con el siguiente pseudocódigo:

```
pred = max(X0, X1);  
if (absCoeffMod == 1 && pred > 0)  
{  
    absCoeff = pred;  
}  
else  
{  
    absCoeff = absCoeffMod – (absCoeffMod <= pred);  
}
```

[0093] El decodificador de video 300 puede determinar, por ejemplo, un valor de nivel predicho para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, recibir un elemento de sintaxis que indica, y en respuesta a que el elemento de sintaxis tiene un valor igual a uno, determinar que un valor de nivel del coeficiente que se está decodificando actualmente es igual al valor de nivel predicho. En otros casos, el decodificador de video 300 puede determinar un valor de nivel predicho para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, recibir un elemento de sintaxis que indica, y en respuesta a que un valor para el elemento de sintaxis es mayor que el valor de nivel predicho, determinar que un valor de nivel del coeficiente que se está decodificando actualmente es igual al valor del elemento de sintaxis más uno. En otros casos, el decodificador de video 300 puede determinar un valor de nivel predicho para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, recibir un elemento de sintaxis que indica, y en respuesta a que un valor para el elemento de sintaxis es menor que el valor de nivel predicho, determinar que un valor de nivel del coeficiente que se está decodificando actualmente es igual al valor del elemento de sintaxis.

[0094] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300

determinan o derivan el contexto del elemento de sintaxis identificado aquí como `abs_level_gt1_flag` en función de si los valores de coeficiente vecino izquierdo y vecino superior son cero. El elemento de sintaxis `abs_level_gt1_flag` es un elemento de sintaxis que se utiliza para codificar valores de coeficientes. El valor de `abs_level_gt1_flag` igual a 1 puede significar, por ejemplo, que el nivel absoluto de un coeficiente es mayor que 1. El valor de `abs_level_gt1_flag` igual a 0 puede, por ejemplo, significar que el nivel absoluto de un coeficiente no es mayor que 1.

[0095] En un ejemplo, el contexto puede ser uno de varios, por ejemplo, tres, de contextos diferentes. Se puede derivar un contexto para el caso en el que tanto un coeficiente vecino izquierdo como un coeficiente vecino superior tengan valores distintos de cero. Se puede derivar otro contexto para el caso en el que solo uno de un coeficiente vecino izquierdo o un coeficiente vecino superior que tenga un valor que no sea cero. Se puede derivar un tercer contexto para el caso en el que tanto un coeficiente vecino izquierdo como un coeficiente vecino superior tengan valores de cero. En un ejemplo, esta derivación de contexto se aplica únicamente a modos que no son BDPCM (modulación de código de impulsos diferenciales de bloque).

[0096] En algunos ejemplos, para escenarios de codificación con un valor vecino no existente o no disponible, como cuando un coeficiente que se está codificando está en el límite izquierdo de un bloque y no hay un vecino izquierdo presente), el codificador de video 200 y el decodificador de video 300 se puede configurar para usar un valor cero al derivar el contexto.

[0097] En algunos ejemplos, la derivación del contexto se puede describir de la siguiente manera:

```
ctxOffset = 0;
if (Exist(left_neighbor) && non-zero(left_neighbor))
{
    ctxOffset += 1;
}
else if (Exist(above_neighbor) && non-zero(above_neighbor))
{
    ctxOffset += 1;
}
```

En tales ejemplos, para un valor vecino no existente/no disponible (por ejemplo, vecino izquierdo de un valor en el límite izquierdo de un bloque), el codificador de video 200 y el decodificador de video 300 pueden configurarse para usar un valor cero para el valor

no disponible al derivar el contexto.

[0098] La FIG. 4 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede realizar las técnicas de la presente divulgación. La FIG. 4 se proporciona con fines explicativos y no debe considerarse una limitación de las técnicas como se ejemplifica y describe ampliamente en la presente divulgación. Para fines de explicación, la presente 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 (H.265) y el estándar de codificación de video VVC (H.266) en desarrollo. Sin embargo, las técnicas de la presente divulgación no se limitan a estos estándares de codificación de video y son aplicables generalmente a la codificación y decodificación de video.

[0099] En el ejemplo de la FIG. 4, 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 transformada 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, la memoria intermedia de imagen decodificada (DPB) 218 y la unidad de codificación por entropía 220. Cualquiera o todos entre 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 transformada 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, la DPB 218 y la unidad de codificación por 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.

[0100] 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 desde, por ejemplo, la fuente de video 104 (FIG. 1). La 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 parte del codificador de video 200. La memoria de datos de video 230 y la DPB 218 pueden estar formadas 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 la 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 integrada en chip con otros componentes del codificador de video 200, como se ilustra, o fuera del chip en relación con esos componentes.

[0101] En la presente 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 desde las diversas unidades del codificador de video 200.

[0102] Las diversas unidades de la FIG. 4 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 unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0103] 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 mediante un *software* ejecutado por los circuitos programables, la memoria 106 (FIG. 1) puede almacenar el

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

[0104] 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.

[0105] 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 de paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM), o similar.

[0106] 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 partición 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.

[0107] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 230 en una serie de CTU y encapsular una o más CTU dentro de un segmento. La unidad de selección de modo 202 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura de QTBT o la estructura de árbol cuaternario de HEVC descrita con anterioridad. Como se describió con anterioridad, el codificador de video 200 puede formar una o más CU 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”.

[0108] 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 la 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 de cuadrados (SSD), la diferencia absoluta media (MAD), diferencias de cuadrados medias (MSD) o similares. La unidad de estimación de movimiento 222 generalmente puede realizar estos cálculos mediante el uso de diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se 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.

[0109] La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV) que definen las posiciones de los bloques de referencia en las imágenes de referencia con respecto a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede proporcionar los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la interpredicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un solo vector de movimiento, mientras que para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de desplazamiento de movimiento 224 puede generar entonces un bloque de predicción mediante los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia mediante el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por los respectivos vectores de movimiento y combinar los datos recuperados, por ejemplo, el promedio de muestra por muestra o el promedio ponderado.

[0110] 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.

[0111] 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. La 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 utilizando la modulación de código de pulso diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse mediante uno o más circuitos sustractores que realizan sustracción binaria.

[0112] 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.

[0113] 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 300 pueden admitir tamaños de CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

[0114] 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 función de una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 a codificar.

[0115] 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.

[0116] La unidad de procesamiento de transformada 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 transformada 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformada 206 puede aplicar una transformada de coseno discreto (DCT), una transformada direccional, una transformada de Karhunen-Loeve (KLT) o una transformada conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformada 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 transformada 206 no aplica transformaciones a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformada 206 no aplica transformaciones a un bloque residual. En tales casos, la unidad de procesamiento de transformada 206 puede generar un bloque de coeficientes, donde los coeficientes corresponden a valores residuales en

lugar de coeficientes de transformación.

[0117] La unidad de cuantificación 208 puede cuantificar coeficientes de transformación en un bloque de coeficiente de transformación, para producir un bloque de coeficiente de transformación cuantificado. Para un bloque codificado en modo de salto de transformación, la unidad de cuantificación 208 puede cuantificar coeficientes en un bloque de coeficientes para producir un bloque de coeficientes cuantificados. La unidad de cuantificación 208 puede cuantificar coeficientes o transformar coeficientes de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, mediante la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado al ajustar el valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información y, por tanto, los coeficientes cuantificados o los coeficientes de transformación pueden tener una precisión menor que los coeficientes originales o los coeficientes de transformación producidos por la unidad de procesamiento de transformada 206.

[0118] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformada inversa 212 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficiente cuantificado, respectivamente, para reconstruir un bloque residual desde el bloque de coeficiente. 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.

[0119] 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 las interferencias de bloqueo a lo largo de los bordes de las CU. Las operaciones de la unidad de filtro 216 pueden omitirse, en algunos ejemplos.

[0120] El codificador de video 200 almacena bloques reconstruidos en la DPB 218. Por ejemplo, en ejemplos donde no se realizan operaciones de la unidad de filtro 216, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en la DPB 218. En ejemplos donde se realizan operaciones de la unidad de filtro 216, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados en la DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia de la DPB 218, formada a partir de los bloques

reconstruidos (y potencialmente filtrados), para interpredecir los bloques de imágenes codificadas posteriormente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en la DPB 218 de una imagen actual para intraprededir otros bloques en la imagen actual.

[0121] En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía bloques de coeficientes cuantificados de la unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para interpredicción o información intramodo para intrapredicción) desde la unidad de selección de modo 202. La unidad de codificación por entropía 220 puede realizar una o más operaciones de codificación por entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 220 puede realizar una operación de codificación de longitud variable 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 por entropía de partición de intervalo de probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de codificación por entropía en los datos. En algunos ejemplos, la unidad de codificación por entropía 220 puede funcionar en modo de derivación donde los elementos de sintaxis no están codificados por entropía.

[0122] 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 una porción o imagen. En particular, la unidad de codificación por entropía 220 puede emitir el flujo de bits.

[0123] 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.

[0124] En algunos ejemplos, las operaciones realizadas con respecto a un bloque de

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

[0125] El codificador de video 200 representa un ejemplo de un dispositivo 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 determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y codificar un valor para el coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado. El primer coeficiente vecino puede ser, por ejemplo, uno de un coeficiente vecino superior o un coeficiente vecino izquierdo, y el segundo coeficiente vecino puede ser el otro coeficiente vecino superior o el coeficiente vecino izquierdo.

[0126] Para codificar el valor del coeficiente que se está codificando actualmente en función del desplazamiento de contexto determinado, el codificador de video 200 puede configurarse para determinar un contexto en función del desplazamiento de contexto determinado, determinar un signo para el coeficiente que se está codificando actualmente y codificar el contexto de uno o más bins de datos en función del contexto determinado para representar el signo del coeficiente que se está codificando actualmente. El codificador de video 200 puede configurarse para cuantificar un valor residual para el bloque residual de datos de video para determinar el valor del coeficiente que se está codificando actualmente. El codificador de video 200 puede configurarse para determinar un bloque de predicción y comparar el bloque de predicción con un bloque original de datos de video para determinar el bloque residual de datos de video.

[0127] Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor

para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para seleccionar el desplazamiento de contexto de tres desplazamientos de contexto disponibles en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino. Los tres desplazamientos de contexto disponibles pueden incluir un primer desplazamiento de contexto para cuando el primer coeficiente vecino y el segundo coeficiente vecino sean ambos iguales a cero o tengan signos opuestos; un segundo desplazamiento de contexto para cuando el primer coeficiente vecino y el segundo coeficiente vecino sean ambos positivos o uno del primer coeficiente vecino y el segundo coeficiente vecino sea igual a cero y el otro del primer coeficiente vecino y el segundo coeficiente vecino sea positivo; y un tercer desplazamiento de contexto para cuando el primer coeficiente vecino y el segundo coeficiente vecino sean ambos negativos o uno del primer coeficiente vecino y el segundo coeficiente vecino sea igual a cero y el otro del primer coeficiente vecino y el segundo coeficiente vecino sea negativo.

[0128] Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es igual a cero y el valor para el segundo coeficiente vecino es igual a cero. Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es uno mayor que cero o menor que cero y el valor para el segundo coeficiente vecino es el otro mayor que cero o menor que cero.

[0129] Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino son ambos valores distintos de cero y con signos opuestos. Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para

establecer el valor de desplazamiento de contexto en un segundo valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es mayor o igual a cero y el valor para el segundo coeficiente vecino es mayor o igual a cero, donde al menos uno del valor para el primer coeficiente vecino o el valor para el segundo coeficiente vecino es mayor o igual a uno.

[0130] Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para establecer el valor de desplazamiento de contexto en un segundo valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino son ambos valores no negativos. Para determinar el desplazamiento de contexto para el coeficiente que se está codificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el codificador de video 200 puede configurarse para establecer el valor de desplazamiento de contexto en un tercer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es menor o igual a cero y el valor para el segundo coeficiente vecino es menor o igual a cero, donde el valor para el primer coeficiente vecino o el valor para el segundo coeficiente vecino es menor o igual a uno negativo (-1). Los valores de desplazamiento primero, segundo y tercero pueden ser, por ejemplo, valores de desplazamiento diferentes. Un desplazamiento de contexto es un valor para determinar un contexto. Por tanto, se puede considerar que los valores de desplazamiento primero, segundo y tercero identifican o hacen referencia a tres contextos diferentes.

[0131] El codificador de video 200 también representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un nivel de coeficiente absoluto para un primer coeficiente vecino de un coeficiente que se está codificando actualmente; determinar un nivel de coeficiente absoluto para un segundo coeficiente vecino del coeficiente que se está codificando actualmente; determinar un nivel de coeficiente absoluto para el coeficiente que se está codificando actualmente; y en función del nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino, codificar uno o más elementos de sintaxis que indiquen el nivel de coeficiente absoluto para el coeficiente que se está

codificando actualmente. El codificador de video 200 puede, por ejemplo, cuantificar un valor residual para el bloque residual de datos de video para determinar el valor del coeficiente que se está codificando actualmente.

[0132] En algunos ejemplos, el codificador de video 200 puede configurarse para determinar un nivel de predictor basado en el nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino y codificar un elemento de sintaxis, con un primer valor para el elemento de sintaxis que indica que el nivel de predictor es igual al nivel de coeficiente absoluto para el coeficiente que se está codificando actualmente y un segundo valor para el elemento de sintaxis que indica que el nivel de predictor no es igual al nivel de coeficiente absoluto para el coeficiente que se está codificando actualmente. En algunos ejemplos, el codificador de video 200 puede estar configurado, por ejemplo, para determinar un nivel de predictor basado en el nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino, determinar un valor para un elemento de sintaxis basado en el nivel de predictor y el nivel de coeficiente absoluto para el coeficiente que se está codificando actualmente, y codificar el elemento de sintaxis. Para determinar el nivel de predictor, el codificador de video 200 puede configurarse para establecer el nivel de predictor igual al mayor entre el nivel de coeficiente absoluto para el primer coeficiente vecino o el nivel de coeficiente absoluto para el segundo coeficiente vecino.

[0133] La FIG. 5 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. 5 se proporciona con fines de explicación y no limita las técnicas como se ejemplifica y describe ampliamente en la presente divulgación. Para fines de explicación, la presente 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.

[0134] En el ejemplo de la FIG. 5, el decodificador de video 300 incluye memoria de memoria intermedia de imagen codificada (CPB) 320, unidad de decodificación por entropía 302, unidad de procesamiento de predicción 304, unidad de cuantificación inversa 306, unidad de procesamiento de transformada inversa 308, unidad de reconstrucción 310, unidad de filtro 312 y memoria intermedia de imagen decodificada (DPB) 314. Cualquiera o todos entre la memoria de CPB 320, la unidad de decodificación por entropía 302, la unidad de procesamiento de predicción 304, la

unidad de cuantificación inversa 306, la unidad de procesamiento de transformada inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y la 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.

[0135] 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 adicionales 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 intrbloque (que puede formar parte de la unidad de desplazamiento de movimiento 316), una unidad afín, una unidad de modelo lineal (LM) o similares. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o componentes funcionales diferentes.

[0136] La memoria de 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 de CPB 320 pueden obtenerse, por ejemplo, del medio legible por computadora 110 (FIG. 1). La memoria de CPB 320 puede incluir una CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. Además, la memoria de CPB 320 puede almacenar datos de video que no sean elementos de sintaxis de una imagen codificada, como datos temporales que representan salidas desde las diversas unidades del decodificador de video 300. La DPB 314 generalmente almacena imágenes decodificadas, que el decodificador de video 300 puede emitir y/o usar como datos de video de referencia al decodificar datos o imágenes posteriores del flujo de bits de video codificado. La memoria de CPB 320 y la DPB 314 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, como DRAM, incluida la SDRAM, la MRAM, la RRAM u otros tipos de dispositivos de memoria. La memoria de CPB 320 y la DPB 314 pueden ser proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de CPB 320 puede estar integrada en chip con otros componentes del decodificador de video 300, o fuera de chip en relación con esos componentes.

[0137] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados desde la memoria 120 (FIG. 1). Es decir, la memoria 120 puede almacenar datos como se explicó con anterioridad con la memoria

de 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 *software* para ser ejecutada por circuitos de procesamiento del decodificador de video 300.

[0138] Las diversas unidades mostradas en la FIG. 5 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. 4, 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 mediante las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o enviar parámetros), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuito distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0139] 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 integrada 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.

[0140] La unidad de decodificación por entropía 302 puede recibir datos de video codificados desde la 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 transformada inversa 308, la unidad de reconstrucción 310 y la unidad de filtro 312 pueden generar datos de video decodificados en función de los elementos de sintaxis extraídos del flujo de bits.

[0141] 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”).

[0142] La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes cuantificados de un bloque de coeficiente 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 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 de forma inversa los coeficientes de transformación. La unidad de cuantificación inversa 306 puede formar así un bloque de coeficiente que incluye coeficientes de transformación.

[0143] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente para un bloque que está transformado, la unidad de procesamiento de transformada 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 transformada 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 de transformación. Para bloques que están codificados en un modo de salto de transformación, la unidad de procesamiento de transformada inversa 308 puede no realizar una transformación inversa y, en estos escenarios de codificación, puede verse como una unidad de paso que no procesa ni altera el bloque de coeficientes.

[0144] 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 por entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es interpredicho, la unidad de compensación de movimiento 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de información de predicción pueden indicar una imagen de referencia en DPB 314 desde la 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 desplazamiento 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. 4).

[0145] 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. 4). La unidad de interpredicción 318 puede recuperar datos de muestras vecinas al bloque actual de la DPB 314.

[0146] La unidad de reconstrucción 310 puede reconstruir el bloque actual mediante el uso del bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0147] 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 las interferencias 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.

[0148] El decodificador de video 300 puede almacenar los bloques reconstruidos en la DPB 314. Por ejemplo, en los ejemplos en los que no se realizan operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en la DPB 314. En los ejemplos donde las operaciones de la unidad de filtro 312 se realizan, la unidad de filtro 312 puede almacenar los bloques reconstruidos filtrados en la DPB 314. Como se discutió con anterioridad, la DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes decodificadas previamente para el desplazamiento de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas desde la DPB para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la FIG. 1.

[0149] 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 configurada para determinar, para un bloque residual de datos de video codificados que usa un modo

de salto de transformación, un valor para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; determinar un valor para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; determinar un desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino; y decodificar un valor para el coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado. El primer coeficiente vecino puede ser, por ejemplo, uno de un coeficiente vecino superior o un coeficiente vecino izquierdo, y el segundo coeficiente vecino puede ser el otro coeficiente vecino superior o el coeficiente vecino izquierdo. Como se describió con anterioridad, para bloques residuales codificados en un modo de salto de transformación, el valor del coeficiente puede corresponder a un valor residual, tal como un valor residual cuantificado o un valor residual no cuantificado.

[0150] Para decodificar el valor del coeficiente que se está decodificando actualmente en función del desplazamiento de contexto determinado, el decodificador de video 300 puede configurarse para determinar un contexto en función del desplazamiento de contexto determinado; recibir uno o más bins de datos; y decodificar contextualmente el uno o más bins de datos en función del contexto determinado para determinar un signo para el coeficiente que se está decodificando actualmente. El decodificador de video 300 puede configurarse para cuantificar de forma inversa el valor del coeficiente que se está decodificando actualmente para determinar un valor residual para el bloque residual de datos de video. El decodificador de video 300 puede configurarse para determinar un bloque residual decodificado en función del valor para el coeficiente que se está decodificando actualmente; añadir el bloque residual decodificado a un bloque de predicción para determinar un bloque reconstruido; realizar una o más operaciones de filtrado en el bloque reconstruido para determinar un bloque decodificado de datos de video; y generar una imagen decodificada de datos de video que incluye el bloque decodificado de datos de video.

[0151] Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para seleccionar el desplazamiento de contexto de tres desplazamientos de contexto disponibles en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino. Los tres desplazamientos de contexto disponibles pueden incluir un primer desplazamiento de contexto para cuando el primer coeficiente

vecino y el segundo coeficiente vecino sean ambos iguales a cero o tengan signos opuestos, un segundo desplazamiento de contexto para cuando el primer coeficiente vecino y el segundo coeficiente vecino sean ambos positivos o uno del primer coeficiente vecino y el segundo coeficiente vecino sea igual a cero y el otro del primer coeficiente vecino y el segundo coeficiente vecino sea positivo, y un tercer desplazamiento de contexto para cuando el primer coeficiente vecino y el segundo coeficiente vecino sean ambos negativo o uno del primer coeficiente vecino y el segundo coeficiente vecino sea igual a cero y el otro del primer coeficiente vecino y el segundo coeficiente vecino sea negativo.

[0152] Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es igual a cero y el valor para el segundo coeficiente vecino es igual a cero. Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es uno mayor que cero o menor que cero y el valor para el segundo coeficiente vecino es el otro mayor que cero o menor que cero. Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un primer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino son valores distintos de cero y tienen signos opuestos.

[0153] Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un segundo valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es mayor o igual a cero y el valor para el segundo coeficiente vecino es mayor o igual a cero, donde al menos uno del valor para el primer coeficiente vecino o el valor para el

segundo coeficiente vecino es mayor o igual a uno. Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un segundo valor de desplazamiento en respuesta a que ambos valores no negativos son el valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino. Para determinar el desplazamiento de contexto para el coeficiente que se está decodificando actualmente en función del valor para el primer coeficiente vecino y el valor para el segundo coeficiente vecino, el decodificador de video 300 puede configurarse para establecer el valor de desplazamiento de contexto en un tercer valor de desplazamiento en respuesta a que el valor para el primer coeficiente vecino es menor o igual a cero y el valor para el segundo coeficiente vecino es menor o igual a cero, en donde el valor para el primer coeficiente vecino o el valor para el segundo coeficiente vecino es menor o igual a uno negativo (-1).

[0154] El decodificador de video 300 también 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 determinar, para un bloque residual de datos de video codificados que usa un modo de salto de transformación, un nivel de coeficiente absoluto para un primer coeficiente vecino de un coeficiente que se está decodificando actualmente; determinar un nivel de coeficiente absoluto para un segundo coeficiente vecino del coeficiente que se está decodificando actualmente; y determinar un nivel de coeficiente absoluto para el coeficiente que se está decodificando actualmente en función del nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino. El decodificador 300 de video puede, por ejemplo, cuantificar de forma inversa el coeficiente que se está decodificando actualmente para determinar un valor residual para el bloque residual de datos de video.

[0155] Para determinar el nivel de coeficiente absoluto para el coeficiente que se está decodificando actualmente, el decodificador de video 300 puede configurarse para determinar un nivel de predictor basado en el nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino y recibir un elemento de sintaxis, con un primer valor para el elemento de sintaxis que indica que el nivel de predictor es igual al nivel de coeficiente absoluto para el coeficiente que se está decodificando actualmente y un segundo valor para el elemento de sintaxis que indica que el nivel de predictor no es igual al nivel de coeficiente absoluto para el

coeficiente que se está decodificando actualmente. Para determinar el nivel de coeficiente absoluto para el coeficiente que se está decodificando actualmente, el decodificador de video 300 puede configurarse para determinar un nivel de predictor basado en el nivel de coeficiente absoluto para el primer coeficiente vecino y el nivel de coeficiente absoluto para el segundo coeficiente vecino, recibir un elemento de sintaxis y determinar el nivel de coeficiente absoluto para el coeficiente que se está decodificando actualmente en función del nivel de predictor y el elemento de sintaxis. Para determinar el nivel del predictor, el decodificador 300 de video puede establecer el nivel del predictor igual al mayor entre el nivel de coeficiente absoluto para el primer coeficiente vecino o el nivel de coeficiente absoluto para el segundo coeficiente vecino.

[0156] Las FIG. 6A y 6B muestran ejemplos de un proceso CABAC en un bin n . En el ejemplo 400 de la FIG. 6A, el intervalo en el bin n incluye RangeMPS y RangeLPS dados por la probabilidad del símbolo menos probable (LPS) (p_σ) dado un cierto estado de contexto (σ). El ejemplo 400 muestra la actualización del intervalo en $n + 1$ cuando el valor de bin n es igual al símbolo más probable (MPS). En este ejemplo, el valor bajo en el intervalo permanece igual, pero el valor del intervalo en el bin $n + 1$ se reduce al valor de RangeMPS en el bin n . El ejemplo 402 de la FIG. 6B muestra la actualización del intervalo en $n + 1$ cuando el valor de bin n no es igual al MPS (es decir, igual al LPS). En este ejemplo, el valor bajo del intervalo se mueve al valor de intervalo más bajo de RangeLPS en el intervalo n . Además, el valor del intervalo en $n + 1$ se reduce al valor de RangeLPS en bin n .

[0157] En un ejemplo del proceso de codificación de video HEVC, el intervalo se expresa con 9 bits y el valor bajo, con 10 bits. Existe un proceso de renormalización para mantener el intervalo y los valores bajos con suficiente precisión. La renormalización ocurre siempre que el intervalo es menor que 256. Por lo tanto, el intervalo siempre es igual o mayor que 256 después de la renormalización. Según los valores de intervalo y el valor bajo, el codificador aritmético binario (BAC) envía al flujo de bits un '0' o un '1' o actualiza una variable interna (llamada BO: bits pendientes) para guardar para salidas futuras. La FIG. 7 muestra ejemplos de salida BAC según el intervalo. Por ejemplo, se envía un "1" al flujo de bits cuando el intervalo y el nivel bajo están por encima de un cierto umbral (por ejemplo, 512). Se envía un "0" al flujo de bits cuando el intervalo y el valor bajo están por debajo de cierto umbral (por ejemplo, 512). No se envía nada al flujo de bits cuando el intervalo y el valor bajo se encuentran entre ciertos umbrales. En su lugar, se puede incrementar el valor de BO y se codifica el siguiente bin.

[0158] En el modelo de contexto CABAC de H.264/AVC y en algunos ejemplos de HEVC, hay 128 estados. Hay 64 probabilidades posibles de LPS (indicadas por el estado σ) que pueden ser de 0 a 63. Cada MPS puede ser cero o uno. Como tal, los 128 estados son 64 probabilidades de estado multiplicadas por los 2 valores posibles para MPS (0 o 1). Por lo tanto, el estado se puede indexar con 7 bits.

[0159] Para reducir el cálculo de los intervalos LPS derivados ($\text{rangeLPS}\sigma$), los resultados para todos los casos pueden precalcularse y almacenarse como aproximaciones en una tabla de consulta. Por lo tanto, el intervalo de LPS se puede obtener sin ninguna multiplicación mediante el uso de una simple búsqueda en una tabla. Evitar la multiplicación puede ser importante para algunos dispositivos o aplicaciones, ya que esta operación puede causar una latencia significativa en muchas arquitecturas de *hardware*.

[0160] Puede usarse una tabla de intervalo de LPS precalculada de 4 columnas en lugar de la multiplicación. El intervalo se divide en cuatro segmentos. El índice de segmento puede derivarse de la pregunta $(\text{range} \gg 6) \& 3$. En efecto, el índice de segmento se deriva desplazando y eliminando bits del intervalo real. La siguiente Tabla 1 muestra los intervalos posibles y sus índices correspondientes.

TABLA 1 - Índice de intervalo

Intervalo	256-319	320-383	384-447	448-511
$(\text{intervalo} \gg 6) \& 3$	0	1	2	3

[0161] La tabla de intervalos de LPS tiene entonces 64 entradas (una para cada estado de probabilidad) multiplicado por 4 (una para cada índice de intervalo). Cada entrada es el intervalo LPS, es decir, el valor de la multiplicación del intervalo por la probabilidad del LPS. En la siguiente Tabla 2 se muestra un ejemplo de parte de esta tabla. La Tabla 2 muestra los estados de probabilidad 9-12. En una propuesta para HEVC, los estados de probabilidad pueden oscilar entre 0 y 63.

TABLA 2 – RangeLPS

Estado Prob (σ)	RangeLPS Índice 0	Índice 1	Índice 2	Índice 3
...	...	...	...	...
9	90	110	130	150
10	85	104	123	142
11	81	99	117	135
12	77	94	111	128
...	...	...	...	...

[0162] En cada segmento (es decir, valor de intervalo), el intervalo de LPS de cada

estado $_{\sigma}$ de probabilidad está predefinido. En otras palabras, el intervalo LPS de un estado $_{\sigma}$ de probabilidad se cuantifica en cuatro valores (es decir, un valor para cada índice de intervalo). El intervalo de LPS específico utilizado en un punto dado depende del intervalo al que pertenece el segmento. El número de posibles intervalos de LPS utilizados en la tabla es una compensación entre el número de columnas de la tabla (es decir, el número de posibles valores de intervalo de LPS) y la precisión del intervalo de LPS. En términos generales, más columnas dan como resultado errores de cuantificación más pequeños de los valores del intervalo de LPS, pero también aumenta la necesidad de más memoria para almacenar la tabla. Menos columnas aumenta los errores de cuantificación, pero también reduce la memoria necesaria para almacenar la tabla.

[0163] Como se describió con anterioridad, cada estado de probabilidad de LPS tiene una probabilidad correspondiente. La probabilidad p para cada estado se deriva de la siguiente manera:

$$p_{\sigma} = \alpha p_{\sigma-1}$$

donde el estado σ es de 0 a 63. La constante α representa la cantidad de cambio de probabilidad entre cada estado de contexto. En un ejemplo, $\alpha = 0,9493$ o, más precisamente, $\alpha = (0,01875/0,5)^{1/63}$. La probabilidad en el estado $\sigma = 0$ es igual a 0,5 (es decir, $p_0 = 1/2$). Es decir, en el estado de contexto 0, el LPS y el MPS son igualmente probables. La probabilidad en cada estado sucesivo se obtiene al multiplicar el estado anterior por α . Como tal, la probabilidad de que el LPS ocurra en el estado de contexto $\alpha = 1$ es $p_0 * 0,9493$ ($0,5 * 0,9493 = 0,47465$). Como tal, a medida que aumenta el índice de estado α , la probabilidad de que ocurra el LPS disminuye.

[0164] La CABAC es adaptativa porque los estados de probabilidad se actualizan para seguir las estadísticas de la señal (es decir, los valores de los intervalos codificados previamente). El proceso de actualización es el siguiente. Para un estado de probabilidad dado, la actualización depende del índice de estado y del valor del símbolo codificado identificado como LPS o MPS. Como resultado del proceso de actualización, se deriva un nuevo estado de probabilidad, que consiste en una estimación de probabilidad de LPS potencialmente modificada y, si es necesario, un valor de MPS modificado.

[0165] En el caso de que un valor bin iguale el MPS, un índice de estado dado puede incrementarse en 1. Esto es para todos los estados excepto cuando ocurre un MPS en el índice de estado 62, donde la probabilidad de LPS ya está en su mínimo (o de manera equivalente, se alcanza la probabilidad máxima de MPS). En este caso, el índice de

estado 62 permanece fijo hasta que se ve un LPS, o se codifica el último valor bin (el estado 63 se usa para el caso especial del último valor bin). Cuando ocurre un LPS, el índice de estado se cambia al disminuir el índice de estado en una cierta cantidad, como se muestra en la siguiente ecuación. Esta regla se aplica en general a cada aparición de un LPS con la siguiente excepción. Suponiendo que un LPS se ha codificado en el estado con índice $\sigma = 0$, que corresponde al caso equiprobable, el índice del estado permanece fijo, pero el valor de MPS se alternará de manera que el valor de LPS y MPS se intercambie. En todos los demás casos, independientemente del símbolo que se haya codificado, el valor de MPS no se modificará. La derivación de las reglas de transición para la probabilidad LPS se basa en la siguiente relación entre una p_{old} de probabilidad LPS dada y su contraparte actualizada p_{new} .

$$p_{new} = \max(\alpha p_{old}, p_{62}) \quad \text{si se produce MPS}$$

$$p_{new} = (1 - \alpha) + \alpha p_{old} \quad \text{si se produce LPS}$$

[0166] Con respecto a una implementación práctica del proceso de estimación de probabilidad en CABAC, es importante señalar que todas las reglas de transición pueden realizarse mediante como máximo dos tablas, cada una de las cuales tiene 63 entradas de valores enteros sin signo de 6 bits. En algunos ejemplos, las transiciones de estado se pueden determinar con una sola tabla TransIdxLPS, que determina, para un índice de estado dado σ , el nuevo índice de estado actualizado TransIdxLPS [σ] en caso de que se haya observado un LPS. Las transiciones controladas por MPS se pueden obtener mediante un incremento simple (saturado) del índice de estado por el valor fijo de 1, lo que da como resultado un índice de estado actualizado mínimo ($\sigma + 1$, 62). La tabla 3 a continuación es un ejemplo de una tabla TransIdxLPS parcial.

TABLA 3 – TransIdxLPS

Estado Prob (σ)	Nuevo estado TransIdxLPS [σ]
...	...
9	6
10	8
11	8
12	8
...	...

[0167] Las técnicas descritas con anterioridad con respecto a las FIG. 6A, 6B y 7 representan simplemente un ejemplo de implementación de CABAC. Debe entenderse que las técnicas de la presente divulgación no se limitan solo a esta implementación descrita de CABAC. Por ejemplo, en enfoques BAC más antiguos (por ejemplo, el enfoque BAC utilizado en H.264/AVC), las tablas RangeLPS y TransIdxLPS se ajustaron

para videos de baja resolución (es decir, formato intermedio común (CIF) y videos cuarto-CIF (QCIF)). Con HEVC y futuros códecs como VVC, una gran cantidad de contenido de video es de alta definición (HD) y, en algunos casos, mayor que HD. El contenido de video con resolución HD o superior a HD tiende a tener estadísticas diferentes a las secuencias QCIF de hace 10 años que se utilizaban para desarrollar H.264/AVC. Como tal, las tablas RangeLPS y TransIdxLPS de H.264/AVC pueden provocar una adaptación entre estados de una manera demasiado rápida. Es decir, las transiciones entre estados de probabilidad, especialmente cuando ocurre un LPS, pueden ser demasiado grandes para el contenido de video en HD más fluido y de mayor resolución. Por lo tanto, los modelos de probabilidad utilizados según las técnicas convencionales pueden no ser tan precisos para contenido en HD y extra HD. Además, como el contenido de video en HD incluye un intervalo mayor de valores de píxeles, las tablas H.264/AVC no incluyen suficientes entradas para tener en cuenta los valores más extremos que pueden estar presentes en el contenido en HD.

[0168] Como tal, para HEVC y para estándares de codificación futuros tales como VVC, las tablas RangeLPS y TransIdxLPS pueden modificarse para tener en cuenta las características de este nuevo contenido. En particular, los procesos BAC para HEVC y los estándares de codificación futuros pueden usar tablas que permitan un proceso de adaptación más lento y pueden dar cuenta de casos más extremos (es decir, probabilidades sesgadas). Por lo tanto, como ejemplo, las tablas RangeLPS y TransIdxLPS pueden modificarse para lograr estos objetivos al incluir más estados e intervalos de probabilidad que los utilizados en BAC con H.264/AVC o HEVC.

[0169] La FIG. 8 es un diagrama de bloques de una unidad de codificación por entropía 220 de ejemplo, por ejemplo, que forma parte del codificador de video 200 como se muestra en la FIG. 4, que puede configurarse para realizar CABAC de acuerdo con las técnicas de la presente divulgación. Se introduce un elemento de sintaxis 418 en la unidad de codificación por entropía 220. Si el elemento de sintaxis ya es un elemento de sintaxis de valor binario (es decir, un elemento de sintaxis que solo tiene un valor de 0 y 1), se puede omitir el paso de binarización. Si el elemento de sintaxis es un elemento de sintaxis con valor no binario (por ejemplo, un elemento de sintaxis representado por varios bits, como niveles de coeficientes), el elemento de sintaxis con valor no binario es binarizado por el binarizador 420. El binarizador 420 realiza una asignación del elemento de sintaxis con valor no binario en una secuencia de decisiones binarias. Estas decisiones binarias a menudo se denominan “bins”. Por ejemplo, para los niveles de coeficiente de transformación, el valor del nivel se puede dividir en bins sucesivos,

donde cada bin indica si el valor absoluto del nivel del coeficiente es mayor que algún valor. Por ejemplo, bin 0 (a veces llamado indicador de significancia) indica si el valor absoluto del nivel del coeficiente es mayor que 0 o no. Bin 1 indica si el valor absoluto del nivel del coeficiente es mayor que 1 o no, y así sucesivamente. Se puede desarrollar una asignación única para cada elemento de sintaxis con valor no binario.

[0170] Cada bin producido por el binarizador 420 se carga al lado de codificación aritmética binaria de la unidad de codificación por entropía 220. Es decir, para un conjunto predeterminado de elementos de sintaxis con valor no binario, cada tipo de bin (por ejemplo, bin 0) se codifica antes que el siguiente tipo de bin (por ejemplo, bin 1). La codificación se puede realizar en modo regular o en modo derivación. En el modo de derivación, el motor de codificación de derivación 426 realiza la codificación aritmética mediante el uso de un modelo de probabilidad fija, por ejemplo, mediante codificación de Golomb-Rice o de Golomb exponencial. El modo de derivación se usa generalmente para elementos de sintaxis más predecibles.

[0171] La codificación en modo regular implica realizar CABAC. CABAC en modo regular sirve para codificar valores bin en los que la probabilidad de un valor bin es predecible dados los valores bin previamente codificado. La probabilidad de que un bin sea un LPS se determina mediante el modelador de contexto 422. El modelador de contexto 422 emite el valor bin y el modelo de contexto (por ejemplo, el estado de probabilidad σ). El modelo de contexto puede ser un modelo de contexto inicial para una serie de bins, o puede determinarse en función de los valores codificados de bins previamente codificados. Como se describió con anterioridad, el modelador de contexto puede actualizar el estado en función de si el bin previamente codificado era un MPS o un LPS.

[0172] Después de que el modelo de contexto y el estado de probabilidad σ sean determinados por el modelador de contexto 422, el motor de codificación regular 424 realiza BAC en el valor bin. De acuerdo con las técnicas de la presente divulgación, el motor de codificación regular 424 realiza BAC mediante el uso de la tabla de TransIdxLPS 430 que incluye más de 64 estados de probabilidad σ . En un ejemplo, la cantidad de estados de probabilidad es 128. TransIdxLPS se utiliza para determinar qué estado de probabilidad se utiliza para un siguiente bin (bin $n + 1$) cuando el bin anterior (bin n) es un LPS. El motor de codificación regular 424 también puede usar una tabla de RangeLPS 428 para determinar el valor de intervalo para un LPS dado un estado de probabilidad particular σ . Sin embargo, de acuerdo con las técnicas de esta descripción, en lugar de utilizar todos los posibles estados de probabilidad σ de la tabla de

TransIdxLPS 430, los índices de estado de probabilidad σ se asignan a índices agrupados para su uso en la tabla de RangeLPS. Es decir, cada índice de la tabla de RangeLPS 428 puede representar dos o más del número total de estados de probabilidad. La asignación del índice de estado de probabilidad σ a índices agrupados puede ser lineal (por ejemplo, al dividirse entre dos), o puede ser no lineal (por ejemplo, una función logarítmica o una tabla de asignación).

[0173] En otros ejemplos de la divulgación, la diferencia entre estados de probabilidad sucesivos puede hacerse más pequeña al establecer el parámetro α para que sea mayor que 0,9493. En un ejemplo, $\alpha = 0,9689$. En otro ejemplo de la divulgación, la probabilidad más alta (p_0) de que ocurra un LPS puede establecerse en un valor inferior a 0,5. En un ejemplo, p_0 puede ser igual a 0,493.

[0174] De acuerdo con una o más técnicas de la presente divulgación, en lugar de usar el mismo valor de una variable utilizada para actualizar un estado de probabilidad en un proceso de codificación aritmética binaria (por ejemplo, uno o más de un tamaño de ventana, un factor de escala (α) y una velocidad de actualización de probabilidad), la unidad de codificación por entropía 220 puede usar diferentes valores de la variable para diferentes modelos de contexto y/o diferentes elementos de sintaxis. Por ejemplo, la unidad de codificación por entropía 220 puede determinar, para un modelo de contexto de una pluralidad de modelos de contexto, un valor de una variable usada para actualizar un estado de probabilidad en un proceso de codificación aritmética binaria y actualizar el estado de probabilidad en función del valor determinado.

[0175] La FIG. 9 es un diagrama de bloques de una unidad de decodificación por entropía 302 de ejemplo, que forma parte del decodificador de video 300 como se muestra en la FIG. 5, que puede configurarse para realizar CABAC de acuerdo con las técnicas de la presente divulgación. La unidad de decodificación por entropía 302 de la FIG. 9 realiza CABAC de una forma inversa a la de la unidad de codificación por entropía 220 descrita en la FIG. 8. Los bits codificados del flujo de bits 448 se introducen en la unidad de decodificación por entropía 302. Los bits codificados se cargan al modelador de contexto 450 o al motor de decodificación de derivación 452 en función de si los bits codificados se codificaron por entropía mediante el uso del modo de derivación o el modo regular. Si los bits codificados se codificaron en modo de derivación, el motor de decodificación de derivación 452 puede, por ejemplo, utilizar la decodificación de Golomb-Rice o de Golomb exponencial para recuperar los elementos de sintaxis con valores binarios o bins de elementos de sintaxis no binarios.

[0176] Si los bits codificados se codificaron en modo regular, el modelador de contexto

450 puede determinar un modelo de probabilidad para los bits codificados y el motor de decodificación regular 454 puede decodificar los bits codificados para producir bins de elementos de sintaxis con valor no binario (o los elementos de sintaxis mismos si tienen valores binarios). Después de que el modelo de contexto y el estado de probabilidad σ sean determinados por el modelador de contexto 450, el motor de decodificación regular 454 realiza BAC en el valor bin. De acuerdo con las técnicas de la presente divulgación, el motor de decodificación regular 454 realiza BAC mediante el uso de la tabla de TransIdxLPS 458 que incluye más de 64 estados de probabilidad σ . En un ejemplo, el número de estados de probabilidad es 128, aunque podrían definirse otros números de estados de probabilidad, de acuerdo con las técnicas de la presente divulgación. La tabla de TransIdxLPS 458 se utiliza para determinar qué estado de probabilidad se utiliza para un siguiente bin (bin $n + 1$) cuando el bin anterior (bin n) es un LPS. El motor de decodificación regular 454 también puede usar una tabla de RangeLPS 456 para determinar el valor de intervalo para un LPS dado un estado de probabilidad particular σ . Sin embargo, de acuerdo con las técnicas de la presente divulgación, en lugar de utilizar todos los estados de probabilidad posibles σ de la tabla de TransIdxLPS 458, los índices de estado de probabilidad σ se asignan a índices agrupados para su uso en la tabla de RangeLPS 456. Es decir, cada índice en la tabla de RangeLPS 456 puede representar dos o más del número total de estados de probabilidad. La asignación del índice de estado de probabilidad σ a índices agrupados puede ser lineal (por ejemplo, al dividir entre dos), o puede ser no lineal (por ejemplo, una función logarítmica o una tabla de asignación).

[0177] En otros ejemplos de la divulgación, la diferencia entre estados de probabilidad sucesivos puede hacerse más pequeña al establecer el parámetro α para que sea mayor que 0,9493. En un ejemplo, $\alpha = 0,9689$. En otro ejemplo de la divulgación, la probabilidad más alta (p_0) de que ocurra un LPS puede establecerse en un valor inferior a 0,5. En un ejemplo, p_0 puede ser igual a 0,493.

[0178] Después de que los bins sean decodificados por el motor de decodificación regular 454, un binarizador inverso 460 puede realizar una asignación inversa para convertir los bins de nuevo en los valores de los elementos de sintaxis con valor no binario.

[0179] La FIG. 10 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 (FIG. 1 y 4), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 10.

[0180] En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual (550). 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 (552). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque original no codificado y el bloque de predicción para el bloque actual. El codificador de video 200 puede entonces transformar y cuantificar los coeficientes del bloque residual (554). En algunos modos de codificación, como un modo de salto de transformación, el codificador de video 200 puede omitir la transformación y solo cuantificar los datos residuales. A continuación, el codificador de video 200 puede escanear los coeficientes cuantificados del bloque residual (556). Durante el escaneo, o después de este, el codificador de video 200 puede codificar por entropía los coeficientes (558). Por ejemplo, el codificador de video 200 puede codificar los coeficientes mediante el uso de CAVLC o CABAC. El codificador de video 200 puede emitir entonces los datos codificados en entropía de los coeficientes (560).

[0181] La FIG. 11 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 5), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 11.

[0182] 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 (570). 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 (572). El decodificador de video 300 puede predecir el bloque actual (574), por ejemplo, mediante 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 de forma inversa los coeficientes reproducidos (576), para crear un bloque de coeficientes cuantificados. El decodificador de video 300 puede entonces cuantificar y transformar de forma inversa los coeficientes para producir un bloque residual (578). En algunos modos de codificación, como el modo de salto de transformación, el decodificador de video 300 puede omitir la transformación inversa y cuantificar únicamente los coeficientes de forma inversa. El decodificador de

video 300 puede finalmente decodificar el bloque actual al combinar el bloque de predicción y el bloque residual (580).

[0183] La FIG. 12 es un diagrama de flujo que ilustra un método de ejemplo para determinar un contexto para codificar (por ejemplo, codificar o decodificar) un signo de un coeficiente de un bloque residual. El bloque residual puede ser, por ejemplo, un bloque residual omitido por transformación. Las técnicas de la FIG. 12 se describirán con respecto a un codificador de video genérico, que puede corresponder a un codificador de video como el codificador de video 200 o un decodificador de video como el decodificador de video 300. Sin embargo, debe entenderse que otros dispositivos pueden estar configurados para realizar un método similar al de la FIG. 12.

[0184] El codificador de video determina un valor (X_0) para un primer coeficiente vecino de un coeficiente que se está codificando actualmente (600). El codificador de video determina un valor (X_1) para un segundo coeficiente vecino del coeficiente que se está codificando actualmente (602). En respuesta a que tanto X_0 como X_1 son iguales a cero (604, sí), el codificador de video establece un contexto para codificar el signo del coeficiente que se está codificando actualmente en un primer contexto (606). En respuesta a que al menos uno de X_0 o X_1 no es igual a cero (604, no), pero X_0 y X_1 tienen signos opuestos (608, sí), el codificador de video también establece un contexto para codificar el signo del coeficiente que se está codificando actualmente al primer contexto (610).

[0185] En respuesta a que al menos uno de X_0 o X_1 no es igual a cero (604, no), X_0 y X_1 no tienen signos opuestos (608, no), y uno de X_0 o X_1 es mayor que cero (612, sí), el codificador de video establece un contexto para codificar el signo del coeficiente que se está codificando actualmente en un segundo contexto, diferente del primer contexto (614). En respuesta a que al menos uno de X_0 o X_1 no es igual a cero (604, no), X_0 y X_1 no tienen signos opuestos (608, no), y ni X_0 ni X_1 son mayores que cero (612, no), el codificador de video establece un contexto para codificar el signo del coeficiente que se está codificando actualmente en un tercer contexto, diferente del primer o segundo contexto (616).

[0186] Debe reconocerse que, según el ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos descritos o eventos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través de procesamiento de subprocesos múltiples, procesamiento de

interrupción o procesadores múltiples, en lugar de secuencialmente.

[0187] 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 la presente divulgación. Un producto de programa de computadora puede incluir un medio legible por computadora.

[0188] 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 que utilice 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, el cable de fibra óptica, el par trenzado, la DSL o las 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 óptico o disco magnético, 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 magnéticos generalmente reproducen datos magnéticamente, mientras que los discos ópticos reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computadora.

[0189] 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 “circuitos 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.

[0190] Las técnicas de la presente divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente 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 las 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 un conjunto de unidades de *hardware* interoperativas, que incluyen uno o más procesadores como se describe con anterioridad, junto con el *software* y/o *firmware* adecuado.

[0191] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.