

FILTRADO DE SEÑAL DE PREDICCIÓN EN INTRAPREDICCIÓN PONDERADA LINEAL AFÍN

[0001] La presente solicitud reivindica la prioridad de la solicitud de patente de EE.UU. N.º **16/868.982**, presentada el 7 de mayo de 2020, que reivindica el beneficio de la solicitud de patente provisional de EE.UU. N.º **62/845.839**, presentada el 9 de mayo de 2019, cuyos contenidos completos se incorporan como referencia en el presente documento.

CAMPO TÉCNICO

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

ANTECEDENTES

[0003] Las capacidades de video digital se pueden incorporar en una amplia gama de dispositivos, incluidos 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, y similares. Los dispositivos de video digital implementan técnicas de codificación de video, como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de video avanzada (AVC), el estándar de codificación de video de alta eficiencia (HEVC), ITU-T H.265/Codificación de video de alta eficiencia (HEVC) y extensiones de dichos estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficiente mediante la implementación de tales técnicas de codificación de video.

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

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

SÍNTESIS

[0005] La presente divulgación describe técnicas que pueden mejorar la intrapredicción, incluida la derivación y señalización de modos para la intrapredicción ponderada lineal, que también puede denominarse intrapredicción matriz o intrapredicción ponderada matriz o intrapredicción ponderada lineal afín (ALWIP). Más específicamente, para un bloque actual de datos de video que está codificado en un modo ALWIP, esta divulgación describe la técnica para filtrar muestras de referencia de límite para generar un bloque de predicción filtrado. El bloque de predicción filtrado puede mejorar la compensación de distorsión de velocidad para los bloques codificados en el modo ALWIP al generar bloques de predicción más precisos.

[0006] Según un ejemplo, un método de decodificación de datos de video incluye determinar que un bloque actual de datos de video está codificado en un modo de intrapredicción ponderada lineal afín (ALWIP); derivar, con base en un conjunto de muestras vecinas del borde izquierdo del bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior, en donde el subconjunto de muestras del borde izquierdo incluye menos muestras que el conjunto de muestras del borde izquierdo y el subconjunto de muestras del borde superior incluye menos muestras que el conjunto de muestras del borde superior; aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; filtrar, usando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y decodificar el bloque actual de datos de video basado en el bloque de predicción final.

[0007] Según 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 que un bloque actual de datos de video está codificado en un modo de intrapredicción ponderada lineal afín (ALWIP); derivar, con base en un conjunto de muestras vecinas del borde izquierdo del

bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior, en el que el subconjunto de muestras del borde izquierdo incluye menos muestras que el conjunto de muestras del borde izquierdo y el subconjunto de muestras del borde superior incluye menos muestras que el conjunto de muestras del borde superior; aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; filtrar, utilizando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y decodificar el bloque actual de datos de video basado en el bloque de predicción final.

[0008] Según otro ejemplo, un medio de almacenamiento legible por computadora que almacena instrucciones que cuando son ejecutadas por uno o más procesadores hacen que el uno o más procesadores determinen que un bloque actual de datos de video está codificado en un modo de intrapredicción ponderada lineal afín (ALWIP)); derivar, con base en un conjunto de muestras vecinas del borde izquierdo del bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior, en el que el subconjunto de muestras del borde izquierdo incluye menos muestras que el conjunto de muestras del borde izquierdo y el subconjunto de muestras del borde superior incluye menos muestras que el conjunto de muestras del borde superior; aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; filtrar, utilizando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y decodificar el bloque actual de datos de video basado en el bloque de predicción final.

[0009] Según otro ejemplo, un aparato para decodificar datos de video incluye medios para determinar que un bloque actual de datos de video está codificado en un modo de intrapredicción ponderada lineal afín (ALWIP); medios para derivar, con base en un conjunto de muestras vecinas del borde izquierdo del bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior, en donde el subconjunto de muestras del borde izquierdo incluye menos muestras que el conjunto de muestras de borde izquierdo y el subconjunto de muestras de borde superior incluye

menos muestras que el conjunto de muestras de borde superior; medios para aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; medios para filtrar, usando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y medios para decodificar el bloque actual de datos de video basado en el bloque de predicción final.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

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

[0013] La FIG. 3 muestra ejemplos de direcciones de intrapredicción, donde las flechas apuntan hacia las muestras de referencia.

[0014] La FIG. 4 muestra un ejemplo de un bloque rectangular de 8x4 donde las muestras de referencia “más cercanas” no se usan para la intrapredicción, pero se pueden usar muestras de referencia más lejanas.

[0015] Las FIG. 5A-5C muestran ejemplos de procesos de mapeo de modos para modos fuera del rango de dirección diagonal.

[0016] La FIG. 6 es un diagrama conceptual que ilustra direcciones de intrapredicción de ejemplo con direcciones de gran angular.

[0017] La FIG. 7A es un diagrama conceptual que ilustra otro ejemplo de direcciones de intrapredicción con direcciones de gran angular.

[0018] La FIG. 7B es una tabla que ilustra la relación entre el modo de intrapredicción y el ángulo de intrapredicción.

[0019] La FIG. 8 es un diagrama conceptual que ilustra divisiones verticales y horizontales de ejemplo de un bloque.

[0020] La FIG. 9 es un diagrama conceptual que ilustra otros ejemplos de divisiones verticales y horizontales de un bloque.

[0021] La FIG. 10 es una ilustración de muestras de referencia de múltiples líneas de referencia que pueden usarse para la intrapredicción del bloque de codificación.

[0022] Las FIGS. 11A y 11B son diagramas conceptuales que ilustran ejemplos de pesos PDPC en modo DC para posiciones de muestra dentro de un bloque de 4X4.

[0023] La FIG. 12 es un diagrama conceptual que ilustra ejemplos de modos angulares de intrapredicción.

[0024] La FIG. 13A es un diagrama conceptual que ilustra un ejemplo de un modo diagonal superior derecho.

[0025] La FIG. 13B es un diagrama conceptual que ilustra un ejemplo de un modo diagonal inferior izquierdo.

[0026] La FIG. 13C es un diagrama conceptual que ilustra un ejemplo de un modo diagonal superior derecho adyacente.

[0027] La FIG. 13D es un diagrama conceptual que ilustra un ejemplo de un modo diagonal izquierdo inferior adyacente.

[0028] La FIG. 14 es un diagrama conceptual que ilustra un ejemplo de un proceso de intrapredicción ponderada lineal afín (ALWIP) en el bloque de 8x8.

[0029] La FIG. 15 ilustra algunos ejemplos de tipos de bandas de límite.

[0030] La FIG. 16 muestra un ejemplo de muestras de referencia de límites.

[0031] La FIG. 17 muestra un ejemplo de derivación para una muestra.

[0032] La FIG. 18 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0033] La FIG. 19 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0034] La FIG. 20 es un diagrama de flujo que ilustra un proceso de codificación de video de ejemplo.

[0035] La FIG. 21 es un diagrama de flujo que ilustra un proceso de decodificación de video de ejemplo.

[0036] La FIG. 22 es un diagrama de flujo que ilustra un proceso de decodificación de video de ejemplo.

DESCRIPCIÓN DETALLADA

[0037] La codificación de video (por ejemplo, codificación de video y/o decodificación de video) típicamente implica predecir un bloque de datos de video a partir de 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 comparando el bloque predictivo, también denominado bloque de predicción, con el bloque original. Por lo tanto, los datos residuales representan una diferencia entre

el bloque predictivo y el bloque de datos de video original, de modo que al agregar los datos residuales al bloque de predicción se obtiene el bloque original de video. En algunos escenarios de codificación, para reducir el número de bits necesarios para señalar los datos residuales, el codificador de video transforma y cuantifica los datos residuales y señala los datos residuales transformados y cuantificados en el flujo de bits codificado. La compresión lograda por los procesos de transformación y cuantificación puede ser con pérdida, lo que significa que los procesos de transformación y cuantificación pueden introducir distorsión en los datos de video decodificados.

[0038] Un decodificador de video decodifica y agrega los datos residuales al bloque predictivo para producir un bloque de video reconstruido que coincida más estrechamente con el bloque de video original 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 artefactos. Un tipo común de artefacto o distorsión se conoce como bloqueo, donde los límites de los bloques utilizados para codificar los datos de video son visibles.

[0039] 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. Como parte de la realización de una o más operaciones de filtrado, el decodificador de video puede realizar, por ejemplo, uno o más filtrados de desbloqueo, filtrados de desplazamiento adaptativo de muestra (SAO) y 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.

[0040] La presente divulgación describe técnicas que pueden mejorar la intrapredicción, incluida la derivación y señalización de modos para la intrapredicción ponderada lineal, que también puede denominarse intrapredicción matriz o intrapredicción ponderada matriz o intrapredicción ponderada lineal afin (ALWIP). Más específicamente, para un bloque actual de datos de video que está codificado en un modo ALWIP, esta divulgación describe la técnica para filtrar muestras de referencia de límite para generar un bloque de predicción filtrado. El bloque de predicción filtrado puede mejorar la compensación de distorsión de velocidad para los bloques codificados en el modo ALWIP al generar bloques de predicción más precisos.

[0041] Como se explica con más detalle a continuación, cuando un codificador de video

codifica un bloque en modo ALWIP, el codificador de video genera un conjunto de muestras “intermedias” predichas multiplicando un número reducido de muestras de límite con una matriz y un vector de sesgo. El codificador de video luego sobremuestrea las muestras intermedias usando interpolación lineal para generar el bloque predicho. Este proceso puede generar errores de predicción que terminan siendo más grandes en los bordes de los bloques de predicción, lo que resulta en valores residuales más grandes que requieren más bits para comprimir los datos de video. Esta divulgación describe técnicas para filtrar las muestras intermedias para generar un bloque de predicción final de una manera que pueda reducir los errores de predicción en el bloque de predicción. Por ejemplo, un codificador de video configurado de acuerdo con las técnicas de esta divulgación puede aplicar un modelo afín a un subconjunto de muestras de borde izquierdo y un subconjunto de muestras de borde superior para generar un bloque intermedio de muestras intermedias y luego filtrar las muestras intermedias aplicando uno o más filtros en dirección vertical utilizando un conjunto completo de muestras de borde izquierdo y un conjunto completo de muestras de borde superior.

[0042] Es decir, las técnicas de esta divulgación pueden dar como resultado un codificador de video, cuando se usa el modo ALWIP, generando un bloque de predicción que coincide más estrechamente con un bloque de datos de video original, y por lo tanto requiere pequeños valores residuales y, por lo tanto, menos bits totales para comprimir. Al usar menos bloques de compresión de bits totales de datos de video codificados en un modo ALWIP, las técnicas de esta divulgación pueden dar como resultado un codificador de video que logre una mejor compensación de distorsión de velocidad.

[0043] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video 100 de ejemplo que puede realizar las técnicas de esta divulgación. Las técnicas de esta 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.

[0044] 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. Cada uno de los dispositivos de origen 102 y dispositivos de destino 116 puede comprender cualquiera de una amplia gama de dispositivos,

incluyendo una computadora de escritorio, computadora portátil (es decir, laptop), dispositivo móvil, tableta, decodificador, un teléfono como un teléfono inteligente, televisión, cámara, dispositivo de visualización, reproductor de medios digitales, consolas de videojuegos, dispositivo de transmisión de video, dispositivos receptores de transmisión o similares. En algunos casos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden estar equipados para la comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[0045] En el ejemplo de la FIG. 1, el dispositivo de origen 102 incluye la fuente de video 104, la memoria 106, el codificador de video 200 y la interfaz de salida 108. El dispositivo de destino 116 incluye la interfaz de entrada 122, el decodificador de video 300, la memoria 120 y el dispositivo de visualización 118. De acuerdo con esta divulgación, el codificador de video 200 del dispositivo de origen 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para codificar un bloque en un modo ALWIP descrito en el presente documento. 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.

[0046] El sistema 100 como se muestra en la FIG. 1 es simplemente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar técnicas para codificar un bloque en un modo ALWIP descrito en el presente documento. El dispositivo de origen 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo de origen 102 genera datos de video codificados para su transmisión al dispositivo de destino 116. Esta divulgación se refiere a un dispositivo de “codificación” como un dispositivo que realiza la codificación (codificación y/o decodificación) de datos. Así, el codificador de video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de video y un decodificador de video, respectivamente. En algunos ejemplos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden funcionar de una manera sustancialmente simétrica de modo que cada uno, el dispositivo de origen 102 y el dispositivo de destino 116, incluye

componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede soportar la transmisión de video unidireccional o bidireccional entre el dispositivo de origen 102 y el dispositivo de destino 116, por ejemplo, para transmisión de video, reproducción de video, transmisión de video o telefonía de video.

[0047] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin codificar) y proporciona una serie secuencial de imágenes (también denominadas “cuadros”) de los datos de video al codificador de video 200, que codifica datos para las imágenes. La fuente de video 104 del dispositivo de origen 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video sin procesar previamente capturado, y/o una interfaz de alimentación de video para recibir el 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.

[0048] La memoria 106 del dispositivo de origen 102 y la memoria 120 del dispositivo de destino 116 representan memorias de uso general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video sin procesar de la fuente de video 104 y datos de video sin decodificar del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de software ejecutables por, por ejemplo, el codificador de video 200 y 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.

[0049] 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 hacia el dispositivo de destino 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para habilitar el dispositivo de origen 102 para transmitir datos de video codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red informática. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video codificados, y la interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un estándar o 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 hacia el dispositivo de destino 116.

[0050] En algunos ejemplos, el dispositivo de origen 102 puede emitir datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a los datos codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados.

[0051] En algunos ejemplos, el dispositivo de origen 102 puede enviar datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar los datos de video codificados 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 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 suscriptor digital (DSL), módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a los datos de video codificados almacenados en el servidor de archivos 114. El servidor de archivos 114 y la interfaz de entrada 122 pueden configurarse para funcionar de acuerdo con un protocolo de transmisión de datos (streaming), un protocolo de transmisión de descarga o una combinación de estos.

[0052] La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableados (por ejemplo, tarjetas Ethernet), componentes de comunicación inalámbrica que funcionan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11 u otros componentes físicos. En los ejemplos en los que la interfaz de salida 108 y la interfaz de entrada 122 comprenden componentes inalámbricos, la interfaz de salida 108 y la interfaz de entrada 122 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con un estándar de comunicación celular, como 4G, 4G LTE (Long-Term Evolution), LTE Advanced, 5G o similar. En algunos ejemplos donde la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con otros estándares inalámbricos, como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee®), un estándar Bluetooth® o similar. En algunos ejemplos, el dispositivo de origen 102 y/o el dispositivo de destino 116 pueden incluir dispositivos respectivos de sistema en un chip (SoC). Por ejemplo, el dispositivo de origen 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al codificador de video 200 y/o la interfaz de salida 108, y el dispositivo de destino 116 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al decodificador de video 300 y/o la interfaz de entrada 122.

[0053] Las técnicas de esta divulgación se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de transmisión por Internet, tales como transmisión dinámica adaptativa sobre HTTP (DASH), video digital que se codifica en

un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

[0054] 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 señalización definida por el codificador de video 200, que también es utilizado por el decodificador de video 300, como elementos de sintaxis que tienen valores que describen características y/o el procesamiento de bloques de video u otras unidades codificadas (por ejemplo, segmentos, imágenes, grupos de imágenes, secuencias o similares). El dispositivo de visualización 118 muestra imágenes decodificadas de los datos de video decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de una variedad de dispositivos de visualización tales como 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.

[0055] 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).

[0056] 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 digitales de señal (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 utilizando uno o más procesadores para realizar las técnicas de esta 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 inalámbrica, tal como un teléfono celular.

[0057] El codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también denominado codificación de video de alta eficiencia (HEVC) o extensiones a estos, 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 patentados o industriales, como ITU-T H.266, también conocido como codificación de video versátil (VVC). Un borrador reciente del estándar VVC se describe en Bross, et al. “Versatile Video Coding (Draft 8),” Joint Video Experts Team (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 17^o encuentro: Bruselas, BE, 7–17 ene. 2020, JVET-Q2001-v15 (en adelante “VVC Draft 8”). Sin embargo, las técnicas de esta divulgación no se limitan a ningún estándar de codificación particular.

[0058] 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 a una representación YUV antes de la codificación, y el decodificador de video 300 convierte la representación YUV al formato RGB. Alternativamente, las unidades de procesamiento previo y posterior (no mostradas) pueden realizar estas conversiones.

[0059] Esta divulgación puede referirse generalmente a la codificación (por ejemplo, codificación y decodificación) de imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. De manera similar, esta divulgación puede referirse a la codificación de bloques de una imagen para incluir el proceso de codificación o decodificación de datos para los bloques, por ejemplo, predicción y/o

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

[0060] La HEVC define varios bloques, que incluyen unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). De acuerdo con la 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 quadtree. Es decir, el codificador de video divide las CTU y CU en cuatro cuadrados iguales y no superpuestos, y cada nodo del quadtree tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse “nodos hoja”, y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de video además puede dividir PU y TU. Por ejemplo, en HEVC, un quadtree residual (RQT) representa la partición de TU. En HEVC, las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU que se intrapredicen incluyen información de intrapredicción, como una indicación de modo interno.

[0061] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para funcionar de acuerdo con la VVC. De acuerdo con la VVC, un codificador de video (como el codificador de video 200) divide una imagen en una pluralidad de unidades de árbol de codificación (CTU). El codificador de video 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol binario quadtree (QTBT) o una estructura de árbol de tipo múltiple (MTT). La estructura QTBT elimina los conceptos de tipos de partición múltiples, como la separación entre CU, PU y TU de la HEVC. Una estructura QTBT incluye dos niveles: un primer nivel dividido de acuerdo con la partición quadtree 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).

[0062] En una estructura de partición MTT, los bloques pueden dividirse usando una partición quadtree (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT) (también llamado árbol ternario (TT)). Una partición de árbol triple o ternario es una partición donde 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.

[0063] 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).

[0064] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar la partición quadtree por HEVC, partición QTBT, partición MTT u otras estructuras de partición. Para fines de explicación, la descripción de las técnicas de esta divulgación se presenta con respecto a la partición QTBT. Sin embargo, debe entenderse que las técnicas de esta divulgación también pueden aplicarse a los codificadores de video configurados para usar la partición quadtree u otros tipos de partición.

[0065] En algunos ejemplos, una CTU incluye un bloque de árbol de codificación (CTB) de muestras de luma, dos CTB correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras, o un CTB de muestras de una imagen monocroma o una imagen que es codificada utilizando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras. Un CTB puede ser un bloque de $N \times N$ de muestras para algún valor de N de modo que la división de un componente en CTB sea una partición. Un componente es una matriz o muestra única de una de las tres matrices (luma y dos croma) que componen una imagen en formato de color 4:2:0, 4:2:2 o 4:4:4 o la matriz o una muestra única de la matriz que compone una imagen en formato monocromo. En algunos ejemplos, un bloque de codificación es un bloque de $M \times N$ de muestras para algunos valores de M y N , de modo que una división de un CTB en bloques de codificación sea una partición.

[0066] 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 mosaicos particular y una fila de mosaicos particular en una imagen. Una columna de mosaicos 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 mosaicos se refiere a una región rectangular de CTU que tiene una altura especificada por elementos de sintaxis (por ejemplo, como en un conjunto de parámetros de imagen) y un ancho igual al ancho de la imagen.

[0067] 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.

[0068] 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 segmento incluye una cantidad de mosaicos completos o solo una secuencia consecutiva de ladrillos completos de un mosaico.

[0069] Esta divulgación puede usar “NxN” y “N por N” indistintamente para referirse a las dimensiones de muestra de un bloque (como un CU u otro bloque de video) en términos de dimensiones verticales y horizontales, por ejemplo, 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 necesariamente deben 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.

[0070] 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.

[0071] Para predecir una CU, el codificador de video 200 generalmente puede formar un bloque de predicción para la CU a través de la interpredicción o la intrapredicción. La interpredicción generalmente se refiere a la predicción de la CU a partir de datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a la predicción de la CU a partir de datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de

predicción usando uno o más vectores de movimiento. El codificador de video 200 generalmente puede realizar una búsqueda de movimiento para identificar un bloque de referencia que coincida estrechamente con la CU, por ejemplo, en términos de diferencias entre la CU y el bloque de referencia. El codificador de video 200 puede calcular una métrica de diferencia utilizando una suma de diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), diferencia absoluta media (MAD), diferencias cuadradas medias (MSD) u otros cálculos de diferencia similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual usando predicción unidireccional o predicción bidireccional.

[0072] Algunos ejemplos de 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 acercar o alejar, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0073] Para realizar la intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. Algunos ejemplos de 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 puede 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 las CTU y CU en orden de exploración de trama (de izquierda a derecha, de arriba a abajo).

[0074] 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 utilizando la predicción avanzada de vectores de movimiento (AMVP) o el modo de fusión. El codificador de video 200 puede usar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afín.

[0075] Después de la predicción, tal como la intrapredicción o la interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para el bloque. Los datos residuales, como un bloque residual, representan diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado utilizando el modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformadas al bloque residual, para producir datos transformados en un dominio de transformada en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada discreta de coseno (DCT), una transformada entera, una transformada wavelet o una transformada conceptualmente similar a los datos de video residuales. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformada, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada Karhunen-Loeve (KLT) o similar. El codificador de video 200 produce coeficientes de transformada después de la aplicación de una o más transformadas.

[0076] Como se indicó anteriormente, después de cualquier transformada para producir coeficientes de transformada, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación generalmente se refiere a un proceso en el que los coeficientes de transformada se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes de transformada, proporcionando una compresión adicional. Al realizar el proceso de cuantificación, el codificador de video 200 puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformada. Por ejemplo, el codificador de video 200 puede redondear para abajo un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación, el codificador de video 200 puede realizar un desplazamiento a la derecha bit a bit del valor a cuantificar.

[0077] Después de la cuantificación, el codificador de video 200 puede explorar los coeficientes de transformada y producir un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. La exploración puede estar diseñada para colocar coeficientes de transformada de energía más alta (y, por lo tanto, de frecuencia más baja) en la parte delantera del vector y para colocar coeficientes de transformada de energía más baja (y, por lo tanto, de frecuencia más alta) en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede utilizar un orden de exploración predefinido para explorar los coeficientes de

transformada cuantificados para producir un vector serializado, y luego codificar por entropía los coeficientes de transformada cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada 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 el decodificador de video 300 en la decodificación de los datos de video.

[0078] Para realizar la 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.

[0079] 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.

[0080] 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.

[0081] 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 usando 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 la información de partición para dividir una imagen en CTU, y dividir 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 además la predicción y la información residual para bloques (por ejemplo, CU) de datos de video.

[0082] La información residual puede estar representada, por ejemplo, por coeficientes de transformada cuantificados. El decodificador de video 300 puede cuantificar y transformar inversamente los coeficientes de transformada cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 usa un modo de predicción señalado (intrapredicción o interpredicción) e información de predicción relacionada (por ejemplo, información de movimiento para interpredicción) para formar un bloque de predicción para el bloque. El decodificador de video 300 puede entonces combinar el bloque de predicción y el bloque residual (muestra por muestra) para reproducir el bloque original. El decodificador de video 300 puede realizar un procesamiento adicional, tal como realizar un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

[0083] De acuerdo con las técnicas de esta divulgación, el decodificador de video 300 puede configurarse para determinar que un bloque actual de datos de video está codificado en un modo ALWIP; derivar, con base en un conjunto de muestras vecinas del borde izquierdo del bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior; aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; filtrar, utilizando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y decodificar el bloque actual de datos de video basado en el bloque de predicción final. El codificador de video 200, como parte de un bucle de decodificación de un proceso de codificación de video, también puede configurarse para determinar que un bloque actual de datos de video esté codificado en un modo ALWIP; derivar, con base en un conjunto de muestras vecinas del borde izquierdo del bloque actual y un conjunto de muestras vecinas del borde superior del bloque actual, un subconjunto de muestras del borde izquierdo y un subconjunto de muestras del borde superior; aplicar un modelo afín al subconjunto de muestras del borde izquierdo y al subconjunto de muestras del borde superior para generar un bloque intermedio de muestras intermedias; filtrar, utilizando el conjunto de muestras vecinas del borde izquierdo y el conjunto de muestras vecinas del borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final; y decodificar el bloque actual de

datos de video basado en el bloque de predicción final.

[0084] Esta divulgación generalmente puede referirse a la “señalización” de 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 anteriormente, el dispositivo de origen 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir al almacenar elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

[0085] Las FIG. 2A y 2B son diagramas conceptuales que ilustran una estructura QTBT 130 de ejemplo, y una CTU 132 correspondiente. Las líneas continuas representan la división quadtree y las líneas de puntos indican la división de árbol binario. En cada nodo de división (es decir, no hoja) del árbol binario, se señala un indicador para indicar qué tipo de división (es decir, horizontal o vertical) se utiliza, donde 0 indica división horizontal y 1 indica división vertical en este ejemplo. Para la división quadtree, no es necesario indicar el tipo de división, ya que los nodos quadtree 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 división de información) para un nivel de árbol de región de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (como la división de información) para un nivel de árbol de predicción de la estructura QTBT 130 (es decir, las líneas de puntos). El codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar datos de video, tales como datos de transformada y predicción, para CU representadas por nodos hoja terminales de la estructura QTBT 130.

[0086] En general, la CTU 132 de la FIG. 2B puede estar asociada con parámetros que definen tamaños de bloques correspondientes a nodos de la estructura QTBT 130 en el primer y segundo nivel. Estos parámetros pueden incluir un tamaño de CTU (que representa un tamaño de CTU 132 en muestras), un tamaño mínimo de quadtree (MinQTSIZE, que representa un tamaño mínimo permitido de nodo hoja quadtree), un tamaño máximo de árbol binario (MaxBTSIZE, que representa un tamaño máximo permitido de nodo raíz de árbol binario), una profundidad máxima del árbol binario (MaxBTDepth, que representa una profundidad máxima permitida de árbol binario) y un

tamaño mínimo de árbol binario (MinBTSsize, que representa el tamaño mínimo permitido del nodo hoja de árbol binario).

[0087] El nodo raíz de una estructura QTBT correspondiente a una CTU puede tener cuatro nodos secundarios en el primer nivel de la estructura QTBT, cada uno de los cuales puede dividirse de acuerdo con la partición quadtree. Es decir, los nodos del primer nivel son nodos hoja (que no tienen nodos secundarios) o tienen cuatro nodos secundarios. El ejemplo de la estructura 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 de árbol binario (MaxBTSsize), entonces los nodos pueden dividirse además 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 de árbol binario (MinBTSsize) o la profundidad máxima permitida de árbol binario (MaxBTDepth). El ejemplo de la estructura QTBT 130 representa los nodos que tienen líneas de puntos 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 transformada, sin ninguna partición adicional. Como se discutió anteriormente, las CU también pueden denominarse “bloques de video” o “bloques”.

[0088] En un ejemplo de la estructura de partición QTBT, el tamaño de la CTU se establece en 128x128 (muestras de luma y dos muestras de croma de 64x64 correspondientes), el MinQTSsize se establece en 16x16, el MaxBTSsize se establece en 64x64, el MinBTSsize (tanto para ancho y altura) se establece en 4, y la MaxBTDepth se establece en 4. La partición quadtree se aplica primero a la CTU para generar nodos hoja quadtree. Los nodos hoja quadtree pueden tener un tamaño de 16x16 (es decir, MinQTSsize) a 128x128 (es decir, el tamaño de la CTU). Si el nodo hoja quadtree es de 128x128, entonces el nodo no se dividirá más por el árbol binario, ya que el tamaño excede el MaxBTS (es decir, 64x64, en este ejemplo). De lo contrario, el árbol binario dividirá el nodo hoja quadtree. Por lo tanto, el nodo hoja quadtree es también el nodo raíz del árbol binario y tiene la profundidad del árbol binario como 0. Cuando la profundidad de árbol binario alcanza la MaxBTDepth (4, en este ejemplo), no se permite más división. Un nodo de árbol binario que tiene un ancho igual al MinBTSsize (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 al MinBTSsize implica que no se permite más división vertical para ese nodo de árbol binario. Como se señaló anteriormente, los

nodos hoja del árbol binario se denominan CU y se procesan más de acuerdo con la predicción y transformada sin más particiones.

[0089] La FIG. 3 muestra ejemplos de direcciones para la intrapredicción, con las flechas apuntando hacia las muestras de referencia. El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar intrapredicciones, usando ángulos tanto amplios como no amplios. Los modos de intrapredicción incluyen el modo de predicción DC, el modo de predicción plana y los modos de predicción direccionales (o angulares). La predicción direccional para bloques cuadrados usa direcciones entre -135 grados y 45 grados del bloque actual en el modelo de prueba VVC 2 (VTM2), J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 2 (VTM2)," 11.º encuentro JVET, Liubliana, SI, julio 2018 (JVET-K1002), como se ilustra en la FIG. 3.

[0090] En VTM2, la estructura de bloque utilizada para especificar el bloque de predicción para la intrapredicción no está restringida a ser cuadrada (ancho w = altura h). Los bloques de predicción rectangulares o no cuadrados ($w > h$ o $w < h$) pueden aumentar la eficiencia de codificación en función de las características del contenido.

[0091] En tales bloques rectangulares, restringir la dirección de la intrapredicción para que esté dentro de -135 grados a 45 grados puede dar como resultado situaciones en las que se usan muestras de referencia más lejanas en lugar de muestras de referencia más cercanas para la intrapredicción. Es probable que dicho diseño tenga un impacto en la eficiencia de la codificación. Puede ser más beneficioso relajar el rango de restricciones para que se puedan usar muestras de referencia más cercanas (más allá del ángulo de -135 a 45 grados) para la predicción. Un ejemplo de tal caso se da en la FIG. 4.

[0092] La FIG. 4 muestra un ejemplo de un bloque rectangular de 8x4 (bloque actual 400) donde las muestras de referencia "más cercanas" (círculo 404) no se usan para intrapredicir el bloque actual 400. En cambio, se pueden usar muestras de referencia más lejanas (círculo 402), debido a una restricción de que la dirección de intrapredicción debe estar en el rango de -135 grados a 45 grados.

[0093] Durante el 12.º encuentro JVET, se propuso una modificación de la intrapredicción de gran angular en L. Zhao, X. Zhao, S. Liu, X. Li, "CE3-related: Unification of angular intra prediction for square and non-square blocks," 12.º encuentro JVET, Macao SAR, CN, oct. 2018, JVET-L0279 y fue adoptada en VTM3. VTM3 se describe en J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 3 (VTM3)," 12.º encuentro JVET, Macao SAR, CN, oct. 2018, JVET-

L1002.

[0094] Esta propuesta incluyó dos modificaciones para unificar la intrapredicción angular para bloques cuadrados y no cuadrados. Primero, las direcciones de predicción angular se modificaron para cubrir direcciones diagonales de todas las formas de bloque. En segundo lugar, todas las direcciones angulares se mantuvieron dentro del rango entre la dirección diagonal inferior izquierda y la dirección diagonal superior derecha para todas las relaciones de aspecto de bloque (cuadrado y no cuadrado) como se ilustra en las FIG. 5A-5C. Además, el número de muestras de referencia en la fila de referencia superior y la columna de referencia izquierda se puede restringir a $2 * \text{ancho} + 1$ y $2 * \text{altura} + 1$ para todas las formas de bloque.

[0095] Las FIG. 5A-5C son diagramas conceptuales que ilustran el mapeo de modo para unidades de codificación con diferentes formas. El codificador de video 200 y el decodificador de video 300 pueden implementar un proceso de mapeo de modo para determinar los modos de intrapredicción disponibles para varias formas y tamaños de CU. La FIG. 5A muestra un bloque cuadrado que no requiere remapeo de modo angular. La FIG. 5B muestra un remapeo de modo angular para un bloque horizontal no cuadrado. La FIG. 5C muestra un remapeo de modo angular para un bloque vertical no cuadrado. En las FIG. 5B y 5C, los modos A y B se reemplazan por los modos mapeados A y B, de modo que todavía hay solo 65 modos angulares disponibles, pero esos 65 modos disponibles son diferentes entre la FIG. 5A, la FIG. 5B y la FIG. 5C.

[0096] En el ejemplo de la FIG. 5A, la CU 502 es un bloque cuadrado (es decir, $w = h$). La dirección diagonal 504 corresponde a un ángulo de predicción de 45 grados, y la dirección diagonal 506 corresponde a un ángulo de predicción de -135 grados. Todos los modos de predicción disponibles para la CU 502 se encuentran entre la dirección diagonal 504 y la dirección diagonal 506 y, por lo tanto, no es necesario remapear el modo.

[0097] En el ejemplo de la FIG. 5B, la CU 512 es un bloque rectangular no cuadrado, donde w es mayor que h . La dirección diagonal 514 representa la dirección diagonal que va desde la esquina inferior izquierda de la CU 512 a la esquina superior derecha de la CU 512, y la dirección diagonal 516 representa la dirección diagonal que va desde la esquina superior derecha de la CU 512 a la esquina inferior izquierda de la CU 512. Como los modos A y B no están entre las direcciones diagonales 514 y 516, los modos A y B se reemplazan por los modos mapeados A y B, de modo que todos los modos de predicción disponibles para la CU 512 están entre la dirección diagonal 514 y la dirección diagonal 516.

[0098] En el ejemplo de la FIG. 5C, LA CU 522 es un bloque rectangular no cuadrado, donde h es mayor que w . La dirección diagonal 524 representa la dirección diagonal que va desde la esquina inferior izquierda de la CU 522 a la esquina superior derecha de la CU 522, y la dirección diagonal 526 representa la dirección diagonal que va desde la esquina superior derecha de la CU 522 a la esquina inferior izquierda de la CU 522. Como los modos A y B no están entre las direcciones diagonales 524 y 526, los modos A y B se reemplazan por los modos mapeados A y B, de modo que todos los modos de predicción disponibles para la CU 522 están entre la dirección diagonal 624 y la dirección diagonal 526.

[0099] La FIG. 6 es una ilustración de los ángulos amplios que se adoptan en VTM2. La FIG. 7A muestra modos de gran angular (etiquetados -1 a -10 y 67 a 76 en la FIG. 6) representados además de los 65 modos angulares. En el ejemplo de la FIG. 7A, el modo 50 corresponde a un ángulo de predicción de -90 grados. El modo 66 corresponde a un ángulo de predicción de -135 grados, y el modo 2 corresponde a un ángulo de predicción de 45 grados.

[0100] La FIG. 7A muestra un ejemplo de ángulos amplios (etiquetados -1 a -14 y 67 a 80 en la FIG. 7A) en VTM3 más allá de los modos 2 y 66 para un total de 93 modos angulares. En el ejemplo de la FIG. 8, el modo 50 corresponde a un ángulo de predicción de -90 grados. El modo 66 corresponde a un ángulo de predicción de -135 grados, y el modo 2 corresponde a un ángulo de predicción de 45 grados. Aunque VTM3 define 95 modos, para cualquier tamaño de bloque solo se permiten 67 modos. Los modos exactos permitidos dependen del ancho del bloque y la relación de altura. Esto se logra restringiendo el rango de modo según el tamaño del bloque.

[0101] La FIG. 7B es una tabla que muestra la relación entre el modo de intrapredicción y el ángulo de intrapredicción. En particular, la Tabla 1 en la FIG. 7B especifica la tabla de mapeo entre el modo de intrapredicción `predModelIntra` y el parámetro angular `intraPredAngle` en VTM3. VTM3 se describe en B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 3)", 12^o encuentro JVET, Macao SAR, CN, oct. 2018, JVET-L100.

[0102] En la Tabla 1, los modos angulares correspondientes a diagonales de bloque no cuadrado se muestran con un símbolo de intercalación (^). Los modos vertical y horizontal se muestran con un signo de número (#) como referencia. Los modos diagonales de bloque cuadrado se muestran en la Tabla 1 con un asterisco (*). A continuación, los modos angulares con un valor `intraPredAngle` positivo se denominan modos angulares positivos (índice de modo <18 o >50), mientras que los modos angulares con un valor `intraPredAngle` negativo se denominan modos angulares

negativos (índice de modo > 18 y < 50).

[0103] El parámetro de ángulo inverso `invAngle` se deriva basado en `intraPredAngle` de la siguiente manera:

$$\text{invAngle} = \text{Round} \left(\frac{256 + 32}{\text{intraPredAngle}} \right) \quad (2-1)$$

[0104] Tener en cuenta que los valores `intraPredAngle` que son múltiplos de 32 (0, 32, 64, 128, 256, 512) siempre se corresponden con la predicción de muestras de matriz de referencia no fraccionarias, como es el caso en la especificación VTM3.

Tabla 2: Modos diagonales correspondientes a varias relaciones de aspecto de bloque.

Relación de aspecto de bloque (ancho/altura)	Modos diagonales
1 (cuadrado)	2, 34, 66
2	8, 28, 72
4	12, 24, 76
8	14, 22, 78
16	16, 20, 80
1/2	-6, 40, 60
1/4	-10, 44, 56
1/8	-12, 46, 54
1/16	-14, 48, 52

[0105] El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar la codificación intrasubpartición (ISP). Se propuso un modo de codificación intrasubpartición (ISP) en S. De Luxán Hernández, H. Schwarz, D. Marpe, T. Wiegand (HHI) “CE3: Line-based intra coding mode” (en adelante, “JVET -L0076”). Al codificar datos de video utilizando el modo de codificación ISP, el codificador de video 200 y el decodificador de video 300 pueden configurarse para dividir (por ejemplo, dividir o partir) bloques intrapredichos de luma vertical u horizontalmente en dos (2) o cuatro (4) subparticiones dependiendo de las dimensiones del tamaño del bloque. A continuación, se describen ejemplos de división de bloques en el modo de codificación de ISP con respecto a la FIG. 8 y la FIG. 9.

[0106] La FIG. 8 es un diagrama conceptual que ilustra ejemplos de divisiones verticales y horizontales de un bloque. Como se muestra en la FIG. 8, el bloque actual 800 es un bloque ISP. Es decir, el bloque 800 es un bloque que se dividirá en subparticiones, y cada una de las subparticiones se codificará mediante intrapredicción. El bloque actual 800 tiene una altura (H) y un ancho. En el modo de codificación ISP, el codificador de video 200 y/o el decodificador de video 300 pueden configurarse para dividir el bloque actual 800 ya sea horizontal o verticalmente. En el ejemplo de la FIG. 8, el codificador

de video 200 y/o el decodificador de video 300 pueden configurarse para dividir el bloque actual 800 en dos subparticiones. Cuando se usa un tipo de división horizontal, el codificador de video 200 y/o el decodificador de video 300 pueden dividir el bloque actual 800 en la subpartición 802 y la subpartición 804. Cada subpartición 802 y subpartición 804 tienen una altura igual a $H/2$ y un ancho igual a W . Cuando se usa un tipo de división vertical, el codificador de video 200 y/o el decodificador de video 300 pueden dividir el bloque actual 800 en la subpartición 806 y la subpartición 808. Cada una de las subparticiones 806 y subpartición 808 tienen una altura igual a H y un ancho igual a $W/2$.

[0107] La FIG. 9 es un diagrama conceptual que ilustra otros ejemplos de divisiones verticales y horizontales de un bloque. La FIG. 9 muestra nuevamente el bloque actual 900, que es un bloque ISP. En este ejemplo, el codificador de video 200 y/o el decodificador de video 300 pueden dividir el bloque actual 900 en cuatro subparticiones. Cuando se usa un tipo de división horizontal, el codificador de video 200 y/o el decodificador de video 300 pueden dividir el bloque actual 900 en la subpartición 910, la subpartición 912, la subpartición 914 y la subpartición 916. Cada una de las subparticiones 910, la subpartición 912, la subpartición 914 y la subpartición 916 tienen una altura igual a $H/4$ y un ancho igual a W . Cuando se usa un tipo de división vertical, el codificador de video 200 y/o el decodificador de video 300 pueden dividir el bloque actual 900 en la subpartición 920, la subpartición 922, la subpartición 924 y la subpartición 926. Cada una de las subparticiones 920, subpartición 922, subpartición 924 y subpartición 926 tiene una altura igual a H y un ancho igual a $W/4$.

[0108] La FIG. 8 y la FIG. 9 son simplemente ejemplos de tipos de división. En otros ejemplos de ISP, un bloque actual puede dividirse en cualquier número de particiones (por ejemplo, 3, 5, 6, etc.). Además, en algunos ejemplos, los tamaños de las subparticiones no necesitan ser simétricos. Es decir, las subparticiones pueden tener diferentes tamaños.

[0109] En un ejemplo, basado en el modo de intracodificación y el tipo de división utilizado, se pueden usar dos clases diferentes de órdenes de procesamiento, que se denominan orden “normal” y orden “invertido”. En el orden normal, la primera subpartición que se procesará es la subpartición que contiene la muestra superior izquierda de la CU y luego continúa hacia abajo (división horizontal) o hacia la derecha (división vertical). El codificador de video 200 puede señalar un bit que indica el tipo de división (por ejemplo, división horizontal o vertical) de la CU al decodificador de video 300. En otro ejemplo, el orden de procesamiento inverso comienza con la subpartición que contiene la muestra inferior izquierda de la CU y continúa hacia arriba, o comienza

con la subpartición que contiene la muestra superior derecha de la CU y continúa hacia la izquierda.

[0110] Una variación de ISP que usa solo el orden de procesamiento normal se usa en JVET WD4. Cabe señalar que los términos subbloque y subparticiones se usan indistintamente en este documento, y ambos se refieren a los bloques obtenidos al dividir un bloque de codificación utilizando ISP.

[0111] A continuación se muestran algunas sintaxis y semánticas asociadas con ISP en JVET WD4, con los símbolos <<*>> y <</**>> que muestran la sintaxis relevante.

Tabla de sintaxis de unidad de codificación

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
...	
} else {	
if(treeType == SINGLE_TREE treeType ==	
DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
<<*>> if (intra_luma_ref_idx[x0][y0] == 0 && <</**>>	
<<*>> (cbWidth <= MaxTbSizeY cbHeight <=	
MaxTbSizeY) && <</**>>	
<<*>> (cbWidth * cbHeight >	
MinTbSizeY * MinTbSizeY) <</**>>	
<<*>>intra_subpartitions_mode_flag[x0][y0] <</**>>	<<*>> ae(v) <</**>>
<<*>> if(intra_subpartitions_mode_flag[x0][y0] == 1	
&& <</**>>	
<<*>> cbWidth <= MaxTbSizeY && cbHeight <=	
MaxTbSizeY) <</**>>	
<<*>>intra_subpartitions_split_flag[x0][y0] <</**>>	<<*>> ae(v) <</**>>
if(intra_luma_ref_idx[x0][y0] == 0 &&	
intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
...	

Tabla de sintaxis del árbol de transformada

	Descriptor
transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	
<<*>>InferTuCbfLuma = 1<</**>>	
<<*>> if(IntraSubPartSplitType == NO_ISP_SPLIT) {<</**>>	
if(tbWidth > MaxTbSizeY tbHeight > MaxTbSizeY) {	
trafoWidth = (tbWidth > MaxTbSizeY) ? (tbWidth/2) : tbWidth	
trafoHeight = (tbHeight > MaxTbSizeY) ? (tbHeight/2) : tbHeight	
transform_tree(x0, y0, trafoWidth, trafoHeight)	
if(tbWidth > MaxTbSizeY)	
transform_tree(x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType)	
if(tbHeight > MaxTbSizeY)	
transform_tree(x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
if(tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY)	

transform_tree(x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
} else {	
transform_unit(x0, y0, tbWidth, tbHeight, treeType, 0)	
}	
<<*>> else if(IntraSubPartitionsSplitType == ISP_HOR_SPLIT) {<<*>>	
<<*>>trafoHeight = tbHeight/NumIntraSubPartitions<<*>>	
<<*>>for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++) <<*>>	
<<*>>transform_unit(x0, y0 + trafoHeight * partIdx, tbWidth, trafoHeight, treeType, partIdx) <<*>>	
<<*>>} else if(IntraSubPartitionsSplitType == ISP_VER_SPLIT) {<<*>>	
<<*>>trafoWidth = tbWidth/NumIntraSubPartitions<<*>>	
<<*>>for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++)	
<<*>>transform_unit(x0 + trafoWidth * partIdx, y0, trafoWidth, tbHeight, treeType, partIdx) <<*>>	
<<*>>}	
}	

Semántica de una unidad de codificación

<<*>>intra_subpartitions_mode_flag [x0] [y0] igual a 1 especifica que la unidad de intracodificación actual se divide en NumIntraSubPartitions [x0] [y0] subparticiones de bloque de transformada rectangular. intra_subpartitions_mode_flag [x0] [y0] igual a 0 especifica que la unidad de intracodificación actual no está dividida en subparticiones de bloque de transformada rectangular.

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

intra_subpartitions_split_flag [x0] [y0] especifica si el tipo de división de intrasubparticiones es horizontal o vertical. Cuando intra_subpartitions_mode_flag [x0] [y0] no está presente, se infiere que es igual a 0.

La variable IntraSubPartitionsSplitType especifica el tipo de división utilizada para el bloque de codificación de luma actual como se ilustra en la Tabla 2-3. IntraSubPartitionsSplitType se deriva de la siguiente manera:

- Si intra_subpartitions_mode_flag [x0] [y0] es igual a 0, IntraSubPartitionsSplitType se establece igual a 0.
- De lo contrario, IntraSubPartitionsSplitType se establece igual a 1 + intra_subpartitions_split_flag [x0] [y0].

Tabla 2-3 - Asociación de nombres a IntraSubPartitionsSplitType

IntraSubPartitionsSplitType	Nombre de IntraSubPartitionsSplitType
0	ISP_NO_SPLIT
1	ISP_HOR_SPLIT
2	ISP_VER_SPLIT

La variable NumIntraSubPartitions especifica el número de subparticiones de

bloques de transformada en las que se divide un bloque de intracodificación de luma. NumIntraSubPartitions se deriva de la siguiente manera:

- Si IntraSubPartitionsSplitType es igual a ISP_NO_SPLIT, NumIntraSubPartitions se establece igual a 1.
- De lo contrario, si una de las siguientes condiciones es verdadera, NumIntraSubPartitions se establece igual a 2:
 - cbWidth es igual a 4 y cbHeight es igual a 8,
 - cbWidth es igual a 8 y cbHeight es igual a 4.
- De lo contrario, NumIntraSubPartitions se establece igual a 4. <</**>>

[0112] La FIG. 10 muestra una ilustración de muestras de referencia de múltiples líneas de referencia que pueden usarse para la intrapredicción del bloque de codificación. El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar predicciones de múltiples líneas de referencia. Las muestras en la vecindad de un bloque de codificación se usan para la intrapredicción del bloque. Típicamente, las líneas de muestra de referencia reconstruidas que están más cercanas a la izquierda y los límites superiores del bloque de codificación se usan como muestras de referencia para la intrapredicción. Sin embargo, la VVC WD4 también permite que otras muestras en la vecindad del bloque de codificación se utilicen como muestras de referencia. La FIG. 10 ilustra las líneas de muestra de referencia que pueden usarse para la intrapredicción. Para cada bloque de codificación, se señala un índice que indica la línea de referencia que se utiliza.

[0113] En la VVC WD4, solo se pueden usar líneas de referencia con MRLIdx igual a 0, 1 y 3. El índice de la línea de referencia utilizado para codificar el bloque (valores 0, 1 y 2 que indican líneas con MRLIdx 0, 1 y 3, respectivamente) se codifica con una palabra de código unaria truncada. Los modos plano y DC no se utilizan para la línea de referencia utilizada que tiene MRLIdx > 0.

[0114] El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar una combinación de intrapredicción dependiente de la posición. La intrapredicción basada en bloques es parte de los estándares de video como AVC, HEVC, VVC, etc. Por lo general, se utilizan líneas de muestras de referencia de bloques reconstruidos adyacentes para predecir muestras dentro del bloque actual. Se pueden usar una o varias líneas de muestras para la predicción. Las muestras de referencia son empleadas por modos de intrapredicción típicos, como los modos DC, plano y angular/direccional.

[0115] Se propuso la combinación de intrapredicción dependiente de la posición

(PDPC) en J. Pfaff, B. Stallenberger, M. Schäfer, P. Merkle, P. Helle, T. Hinz, H. Schwarz, D. Marpe, T. Wiegand (HHI) “CE3: Affine linear weighted intra prediction (CE3-4.1, CE3-4.2)” (JVET-N0217) y más simplificada en JVET-M0102. En J. Chen, Y. Ye, S. H. Kim, “Algorithm description for Versatile Video Coding and Test Model 3 (VTM3)” (JVET-L1002), Macao, CN, oct 2018, presentado a la convocatoria de propuestas de JVET, se aplica PDPC a los modos plano, DC, horizontal y vertical sin señalización como se resume a continuación. En F. Bossen, K. Misra, “Non-CE3: A unified luma intra mode list construction process” (JVET-M0528), la PDPC se amplió aún más a modos direccionales diagonales y modos adyacentes a modos direccionales diagonales.

[0116] La muestra de predicción $pred(x,y)$ ubicada en (x, y) se predice con un modo de intrapredicción (DC, plano, angular) y su valor se modifica usando la expresión PDPC para una sola línea de muestra de referencia:

$$pred(x,y) = (wL \times R_{x-1,y} + wT \times R_{x,-1} - wTL \times R_{-1,-1} + (64 - wL - wT + wTL) \times pred(x,y) + 32) \gg 6, \text{ (Ec. 1)}$$

donde $R_{x-1,y}$, $R_{-1,y}$ representan las muestras de referencia ubicadas en la parte superior e izquierda de la muestra actual (x, y) , respectivamente, y $R_{-1,-1}$ representa la muestra de referencia ubicada en la esquina superior izquierda del bloque actual. Para el modo DC, los pesos se calculan de la siguiente manera para un bloque con dimensiones de *ancho* y *alto*:

$$wT = 32 \gg ((y < 1) \gg shift), wL = 32 \gg ((x < 1) \gg shift), wTL = (wL \gg 4) + (wT \gg 4),$$

con $shift = (\log_2(width) + \log_2(height) + 2) \gg 2$,

mientras que para el modo plano $wTL = 0$, para el modo horizontal $wTL = wT$ y para el modo vertical $wTL = wL$. Los pesos de PDPC se pueden calcular solo con adiciones y desplazamientos. El valor de $pred(x,y)$ se puede calcular en un solo paso utilizando la Ec. 1.

[0117] La FIG. 11A ilustra los pesos PDPC en modo DC (wL , wT , wTL) para la posición $(0, 0)$ dentro de un bloque de 4×4 . La FIG. 11B ilustra los pesos PDPC en modo DC (wL , wT , wTL) para la posición $(1, 0)$ dentro de un bloque de 4×4 . Si se aplica PDPC a los intramodos DC, plano, horizontal y vertical, no se aplican filtros de límite adicionales, como el filtro de límite de modo DC o los filtros de borde de modo horizontal/vertical. La Ecuación 1 puede generalizarse para incluir líneas de muestra de referencia adicionales (por ejemplo, sin limitarse a muestras una fila arriba o una fila a la izquierda del bloque actual). En este caso, hay múltiples muestras de referencia disponibles en las vecindades de $R_{x,-1}$, $R_{-1,y}$, $R_{-1,-1}$ y cada una puede tener un peso asignado que puede

optimizarse, por ejemplo, mediante entrenamiento.

[0118] Las técnicas descritas en la solicitud de patente U. S. 16/371.638, presentada el 1 de abril de 2019, extienden PDPC a los intramodos diagonales y a los modos angulares adyacentes a los modos diagonales. Los intramodos diagonales previstos son los modos que predicen de acuerdo con las direcciones inferior izquierda y superior derecha, así como varios modos angulares adyacentes, por ejemplo, N modos adyacentes entre el modo diagonal inferior izquierdo y el modo vertical, y N o M modos adyacentes entre el modo diagonal superior derecho y el modo horizontal. La FIG. 12 ilustra la identificación de los modos angulares. En general, los modos adyacentes pueden ser un subconjunto seleccionado de modos angulares disponibles. El espacio entre modos angulares puede ser no uniforme y algunos modos angulares pueden omitirse.

[0119] Las FIG. 13A–13D ilustran la definición de muestras utilizadas por la extensión PDPC a intramodos angulares diagonales y adyacentes. La FIG. 13A ilustra la definición de las muestras de referencia $R_{x,-1}$, $R_{-1,y}$ y $R_{-1,-1}$ para la extensión de PDPC al modo diagonal superior derecho. La muestra de predicción $\text{pred}(x', y')$ se encuentra en (x', y') dentro del bloque de predicción. La coordenada x de la muestra de referencia $R_{x,-1}$ viene dada por: $x = x' + y' + 1$ y la coordenada y de la muestra de referencia $R_{-1,y}$ también está dada por: $y = x' + y' + 1$. Los pesos PDPC para el modo diagonal superior derecho son, por ejemplo: $w_T = 16 \gg ((y' < 1) \gg \text{shift})$, $w_L = 16 \gg ((x' < 1) \gg \text{shift})$, $w_{TL} = 0$.

[0120] De forma similar, la FIG. 13B ilustra la definición de las muestras de referencia R_x , -1 , $R_{-1,y}$ y $R_{-1,-1}$ para la extensión de PDPC al modo diagonal inferior izquierdo. La coordenada x de la muestra de referencia R_x , -1 viene dada por: $x = x' + y' + 1$, y la coordenada y de la muestra de referencia $R_{-1,y}$ es: $y = x' + y' + 1$. Los pesos de PDPC para el modo diagonal superior derecho son, por ejemplo: $w_T = 16 \gg ((y' < 1) \gg \text{shift})$, $w_L = 16 \gg ((x' < 1) \gg \text{shift})$, $w_{TL} = 0$.

[0121] En las FIG. 13A y 13B, el codificador de video 200 y el decodificador de video 300 pueden determinar cada uno una fila que está por encima del bloque actual (por ejemplo, inmediatamente encima, pero las técnicas no están tan limitadas) y determinar una coordenada x en la fila determinada. La coordenada x en la fila determinada es igual a una coordenada x de la muestra de predicción más una coordenada y de la muestra de predicción más 1. El codificador de video 200 y el decodificador de video 300 pueden determinar una muestra de referencia de la una o más muestras de referencia con base en la fila determinada y la coordenada x determinada.

[0122] De manera similar, en las FIG. 13A y 13B, el codificador de video 200 y el decodificador de video 300 pueden determinar una columna que queda a la izquierda del bloque actual (por ejemplo, inmediatamente a la izquierda, pero las técnicas no están tan limitadas) y determinar una coordenada y en la columna determinada. La coordenada y en la columna determinada es igual a una coordenada x de la muestra de predicción más una coordenada y de la muestra de predicción más 1. El codificador de video 200 y el decodificador de video 300 pueden determinar una muestra de referencia de la una o más muestras de referencia con base en la columna determinada y la coordenada y determinada.

[0123] Con base en las coordenadas x e y determinadas, el codificador de video 200 y el decodificador de video 300 pueden determinar las muestras de referencia (por ejemplo, una primera muestra de referencia basada en la fila determinada y la coordenada x determinada y una segunda muestra de referencia basada en la columna determinada y la coordenada y determinada). Además, el codificador de video 200 y el decodificador de video 300 pueden determinar los pesos de acuerdo con las técnicas de ejemplo anteriores para los modos diagonales (por ejemplo, modo diagonal superior derecho y modo diagonal inferior izquierdo, como dos ejemplos). Luego, basándose en la Ecuación 1 (como un ejemplo no limitativo), el codificador de video 200 y el decodificador de video 300 pueden determinar la muestra de predicción modificada (por ejemplo, $\text{pred}(x,y)$).

[0124] El caso de un modo diagonal superior derecho adyacente se ilustra en la FIG. 13C. En general, para el ángulo α definido en la FIG. 3, la coordenada y de la muestra de referencia R-1,y se determina de la siguiente manera: $y = y' + \tan(\alpha) \times (x' + 1)$ y la coordenada x de Rx,-1 viene dada por: $x = x' + \cotan(\alpha) \times (y' + 1)$, con $\tan(\alpha)$ y $\cotan(\alpha)$ la tangente y la cotangente del ángulo α . Los pesos de PDPC para un modo diagonal superior derecho adyacente son, por ejemplo: $wT = 32 \gg ((y' < 1) \gg \text{shift})$, $wL = 32 \gg ((x' < 1) \gg \text{shift})$, $wTL = 0$ or $wT = 32 \gg ((y' < 1) \gg \text{shift})$, $wL = 0$, $wTL = 0$.

[0125] De manera similar, el caso de un modo diagonal inferior izquierdo adyacente se ilustra en la FIG. 13D. En general, para el ángulo β definido en la FIG. 3, la coordenada x de la muestra de referencia Rx,-1 se determina de la siguiente manera $x = x' + \tan(\beta) \times (y' + 1)$ y la coordenada y de R-1,y viene dada por $y = y' + \cotan(\beta) \times (x' + 1)$, con $\tan(\beta)$ y $\cotan(\beta)$ la tangente y la cotangente del ángulo β . Los pesos PDPC para un modo diagonal inferior izquierdo adyacente son, por ejemplo: $wL = 32 \gg ((x' < 1) \gg \text{shift})$, $wT = 32 \gg ((y' < 1) \gg \text{shift})$, $wTL = 0$ or $wL = 32 \gg ((x' < 1) \gg \text{shift})$, $wT = 0$, $wTL = 0$.

[0126] En las FIG. 13C y 13D, el codificador de video 200 y el decodificador de video 300 pueden determinar cada uno una fila que está por encima del bloque actual (por ejemplo, inmediatamente encima, pero las técnicas no están tan limitadas) y determinar una coordenada x en la fila determinada. La coordenada x en la fila determinada se basa en un ángulo del modo de intrapredicción angular. El codificador de video 200 y el decodificador de video 300 pueden determinar una muestra de referencia de la una o más muestras de referencia con base en la fila determinada y la coordenada x determinada.

[0127] Para determinar la coordenada x en la fila determinada, el codificador de video 200 y el decodificador de video 300 pueden determinar una de una cotangente (por ejemplo, para el modo diagonal superior derecho adyacente) o tangente (por ejemplo, para el modo diagonal inferior izquierdo adyacente) del ángulo del modo de intrapredicción angular. El codificador de video 200 y el decodificador de video 300 pueden determinar la coordenada x en la fila determinada basándose en una de las cotangente o tangente del ángulo del modo de intrapredicción angular, una coordenada x de la muestra de predicción y una coordenada y de la muestra de predicción. Por ejemplo, para el modo de intrapredicción angular diagonal superior derecho adyacente, la coordenada x en la fila determinada es igual a $x' + \cotan(\alpha) \times (y' + 1)$ y para el modo diagonal inferior izquierdo adyacente, la coordenada x en la fila determinada es igual a $x' + \tan(\beta) \times (y' + 1)$, donde x' e y' son las coordenadas x e y de la muestra de predicción que se está modificando.

[0128] De manera similar, en las FIG. 13C y 13D, el codificador de video 200 y el decodificador de video 300 pueden determinar una columna que queda a la izquierda del bloque actual (por ejemplo, inmediatamente a la izquierda, pero las técnicas no están tan limitadas) y determinar una coordenada y en la columna determinada. La coordenada y en la columna determinada se basa en un ángulo del modo de intrapredicción angular. El codificador de video 200 y el decodificador de video 300 pueden determinar una muestra de referencia de la una o más muestras de referencia con base en la columna determinada y la coordenada y determinada.

[0129] Para determinar la coordenada y en la columna determinada, el codificador de video 200 y el decodificador de video 300 pueden determinar uno de una cotangente (por ejemplo, para el modo diagonal inferior izquierdo adyacente) o tangente (por ejemplo, para el modo diagonal superior derecho adyacente) del ángulo del modo de intrapredicción angular. El codificador de video 200 y el decodificador de video 300 pueden determinar la coordenada y en la columna determinada basándose en una de

la cotangente o tangente del ángulo del modo de intrapredicción angular, una coordenada x de la muestra de predicción y una coordenada y de la muestra de predicción. Por ejemplo, para el modo de intrapredicción angular diagonal superior derecho adyacente, la coordenada y en la columna determinada es igual a $y' + \tan(\alpha) \times (x' + 1)$ y para el modo diagonal inferior izquierdo adyacente, la coordenada y en la columna determinada es igual a $y' + \cotan(\beta) \times (x' + 1)$, donde x' e y' son las coordenadas x e y de la muestra de predicción que se está modificando.

[0130] Con base en las coordenadas x e y determinadas, el codificador de video 200 y el decodificador de video 300 pueden determinar las muestras de referencia (por ejemplo, una primera muestra de referencia basada en la fila determinada y la coordenada x determinada y una segunda muestra de referencia basada en el columna determinada y coordenada y determinada). Además, el codificador de video 200 y el decodificador de video 300 pueden determinar los pesos de acuerdo con las técnicas de ejemplo anteriores para los modos diagonales adyacentes (por ejemplo, modo diagonal superior derecho adyacente y modo diagonal inferior izquierdo adyacente, como dos ejemplos). Luego, basándose en la Ecuación 1 (como un ejemplo no limitativo), el codificador de video 200 y el decodificador de video 300 pueden determinar la muestra de predicción modificada (por ejemplo, $\text{pred}(x,y)$).

[0131] Lo anterior describe técnicas de ejemplo para los modos diagonales superior derecho e inferior izquierdo y los modos diagonales superior derecho adyacente e inferior izquierdo adyacente como modos angulares de ejemplo para los que se puede aplicar PDPC. Las técnicas de ejemplo también pueden extenderse a otros modos angulares. Además, en algunos ejemplos, la una o más muestras de referencia tienen una coordenada x e y que es diferente de una coordenada x e y de la muestra de predicción en el bloque de predicción. Por ejemplo, en las ecuaciones de ejemplo anteriores para determinar las coordenadas x e y en filas y columnas respectivas para determinar las muestras de referencia, la coordenada x es diferente de la coordenada x de la muestra de predicción que se está modificando y la coordenada y es diferente de la coordenada y de la muestra de predicción que se está modificando. Es decir, las muestras de referencia pueden no estar en la misma fila o la misma columna que la muestra de predicción que se está modificando.

[0132] Como es el caso para PDPC de modo DC, plano, horizontal y vertical, no hay filtrado de límite adicional, por ejemplo como se especifica en J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, "Algorithm description of Joint Exploration Test Model 7," 7.º encuentro JVET, Torino, Italia, julio 2017, JVET-G1001, para modos diagonales

adyacentes y diagonales cuando PDPC se extiende a estos modos angulares.

[0133] El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar ALWIP. Es decir, el codificador de video 200 y el decodificador de video 300 pueden configurarse para codificar y decodificar bloques de datos de video en un modo ALWIP. ALWIP como se describe en JVET-N0217 genera una predicción de un bloque a partir de las muestras de referencia vecinas utilizando un modelo de predicción ponderado lineal afín. Las muestras vecinas se procesan primero. En algunos casos, las muestras vecinas se submuestrean y luego se utilizan para obtener (usando el modelo afín) un conjunto de muestras reducidas que se asemeja a una versión submuestreada intermedia de las muestras predichas. La predicción final se obtiene mediante el sobremuestreo (según sea necesario) de los valores intermedios.

[0134] Una ilustración del proceso ALWIP se da en la FIG. 14. La FIG. 14 muestra un ejemplo de proceso ALWIP para un bloque de 8x8. Las muestras de límite 1402 representan muestras vecinas en el límite del bloque de 8x8 e incluyen muestras de límite superior ($bdry_{top}$) sobre el bloque de 8x8, y muestras de límite izquierdo ($bdry_{left}$) a la izquierda del bloque de 8x8. El codificador de video 200 o decodificador de video 300 submuestra muestras de límite 1402 para obtener muestras de límite *reducido* 1404, que incluyen muestras de límite superior reducido ($bdry_{red}^{top}$) y muestras de límite izquierdo reducido ($bdry_{red}^{left}$). El codificador de video 200 o el decodificador de video 300 multiplican una representación vectorial de las muestras de límite, $bdry_{red}$, con una matriz A_k y agregan un término de desplazamiento/sesgo b_k para obtener una versión submuestreada del bloque predicho, $pred_{red}$, que está representado por las muestras grises dentro del bloque 1406. El codificador de video 200 o el decodificador de video 300 obtiene un bloque de predicción final 1408 mediante el sobremuestreo de las muestras predichas $pred_{red}$ junto con las muestras de límite para determinar los valores para las otras muestras, es decir, las muestras blancas, en el bloque 1406. La matriz A_k y un el desplazamiento, o sesgo, vector b_k se eligen en función del valor de modo indicado para el bloque.

[0135] Una ilustración del proceso ALWIP se da en la FIG. 11. El proceso ALWIP de la FIG. 11 puede realizarse mediante el codificador de video 200 y el decodificador de video 300. Las muestras de referencia del bloque (también denominadas muestras de límite) se submuestrean para obtener muestras de límite reducidas. La representación vectorial de las muestras de límite, $bdry_{red}$, se multiplica con una matriz A_k y se agrega un término de desplazamiento/sesgo b_k para obtener una versión submuestreada del bloque predicho, $pred_{red}$. La predicción final se obtiene sobremuestreando estas

muestras predichas $pred_{red}$ junto con las muestras de límite. La matriz A_k y un vector de desplazamiento/sesgo b_k se eligen en función de un valor de modo indicado para el bloque. Una combinación de una matriz A_k y un vector de desplazamiento/sesgo b_k puede denominarse en este documento “modo ALWIP”.

[0136] Para derivar las muestras predichas intermedias, el codificador de video 200 y el decodificador de video 300 usan un modelo de predicción ponderado lineal afín. Se definen tres tipos. El número de muestras intermedias derivadas difiere para cada tipo de la siguiente manera:

- 1) 4x4 para tamaños de bloque de ancho y alto iguales a 4
- 2) 8x8 para tamaños de bloque de ancho y alto tanto menores que iguales a 8, excepto cuando el ancho y alto son iguales a 4 (es decir, bloques de 4x8, 8x4 y 8x8)
- 3) 16x16 para bloques donde al menos uno de ancho y alto es mayor que 8.

En cada uno de estos tres casos, se utiliza un número diferente de modos ALWIP: 35, 19 y 11, respectivamente.

[0137] El codificador de video 200 y el decodificador de video 300 pueden configurarse para señalar el ALWIP de la siguiente manera:

- a) Se señala un indicador (`alwip_flag`) para indicar que el bloque actual está codificado con ALWIP.
- b) Cuando el bloque está codificado con ALWIP, se señala otro indicador para indicar si el bloque actual está codificado con un modo ALWIP-MPM o no.
 - a. Si el bloque actual está codificado con el ALWIP MPM, entonces se indica el índice MPM.
 - b. De lo contrario, se indica un índice al valor del modo restante.

El `alwip_flag` puede estar codificado en contexto con cuatro contextos permitidos:

- Si el ancho del bloque $> 2 * \text{alto}$ o el alto $> 2 * \text{ancho}$, se utiliza el contexto 3.
- De lo contrario, se usa el contexto `ctxId`, donde `ctxId` se deriva de la siguiente manera:
 - o Inicializado `ctxId` a 0
 - o Si el bloque vecino izquierdo está codificado con ALWIP, `ctxId++`
 - o Si el bloque vecino que está encima está codificado con ALWIP, `ctxId++`

[0138] El codificador de video 200 y el decodificador de video 300 pueden configurarse para derivar el ALWIP MPM de la siguiente manera:

- 1) `LeftIntraMode` y `AboveIntraMode` se inicializan a -1
- 2) Si el bloque vecino izquierdo está intracodificado

- a. Si el bloque vecino izquierdo está codificado con el modo ALWIP L
 - i. Si L es del mismo tipo ALWIP que el bloque actual, entonces LeftIntraMode se establece igual a L .
 - b. El intramodo del bloque vecino izquierdo se mapea a un modo ALWIP del mismo tipo que el bloque actual y se asigna a LeftIntraMode.
- 3) Si el bloque vecino que está encima está intracodificado:
- a. Si el bloque vecino que está encima está codificado con el modo ALWIP A
 - i. Si A es del mismo tipo ALWIP que el bloque actual, entonces AboveIntraMode se establece igual a A .
 - b. El intramodo del bloque vecino que está encima se mapea a un modo ALWIP del mismo tipo que el bloque actual y se asigna a AboveIntraMode.
- 4) Los MPM se derivan en función de LeftIntraMode y AboveIntraMode.

[0139] En esta divulgación, los bloques codificados con ALWIP pueden denominarse bloques codificados con ALWIP o bloques ALWIP; otros bloques (codificados con intrapredicción regular, intrasubparticiones o múltiples líneas de referencia) pueden denominarse bloques no ALWIP

[0140] El codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar una interpolación lineal de un solo paso. Para un bloque $W \times H$ con $\max(W, H) \geq 8$, la señal de predicción surge de la señal de predicción reducida $pred_{red}$ en $W_{red} \times H_{red}$ por interpolación lineal. Dependiendo de la forma del bloque, el codificador de video 200 y el decodificador de video 300 realizan una interpolación lineal en dirección vertical, horizontal o en ambas direcciones. En algunos ejemplos, si la interpolación lineal se aplica en ambas direcciones, entonces el codificador de video 200 y el decodificador de video 300 aplican primero la interpolación lineal en una dirección horizontal si $W < H$ o primero en la dirección vertical.

[0141] Considerar sin pérdida de generalidad un bloque $W \times H$ con $\max(W, H) \geq 8$ y $W \geq H$. Entonces, el codificador de video 200 y el decodificador de video 300 pueden realizar la interpolación lineal unidimensional como sigue. Para fines de explicación, se describirá la interpolación lineal con respecto a una dirección vertical. Primero, la señal de predicción reducida se extiende hacia arriba por la señal de límite. Definir el factor de sobremuestreo vertical $U_{ver} = H/H_{red}$ y escribir $U_{ver} = 2^{u_{ver}} > 1$. Luego, definir la señal de predicción reducida extendida por

$$pred_{red}[x][-1] = \begin{cases} bdry_{red}^{top}[x] & \text{para } W = 8 \\ bdry_{redll}^{top}[x] & \text{para } W > 8. \end{cases}$$

[0142] Entonces, a partir de esta señal de predicción reducida extendida, la señal de predicción lineal interpolada verticalmente es generada por

$$\begin{aligned} pred_{red}^{ups,ver}[x][U_{ver} \cdot y + k] \\ = \left((U_{ver} - k - 1) \cdot pred_{red}[x][y - 1] + (k + 1) \cdot pred_{red}[x][y] + \frac{U_{ver}}{2} \right) \\ \gg u_{ver} \end{aligned}$$

para $0 \leq x < W_{red}$, $0 \leq y < H_{red}$ y $0 \leq k < U_{ver}$.

[0143] Las técnicas descritas anteriormente incluyen algunos problemas potenciales. ALWIP genera un conjunto de muestras predichas “intermedias” al multiplicar las muestras de límite reducidas con una matriz y un vector de sesgo. Las muestras intermedias luego se sobremuestran usando interpolación lineal según sea necesario para generar el bloque predicho. Aunque la matriz utilizada para ALWIP se elige de un conjunto de varias matrices, cualquier conjunto finito de matrices puede no predecir (y en algunos casos puede ser imposible) eficientemente bloques casi innumerables que ocurren en el contenido de video. Los errores de predicción pueden ser mayores en los bordes de los bloques de predicción, lo que resulta en más bits necesarios para comprimir. A pesar de la interpolación lineal de las muestras usando el bloque de predicción intermedio y las muestras de límite para generar las muestras restantes, no todas las muestras de límite se utilizan para la función de interpolación. Como se muestra en la FIG. 14, el bloque predicho final *pred* se genera a partir de las muestras predichas intermedias, un límite que se submuestra (la parte superior en el ejemplo de la FIG. 14) y un límite sin modificaciones. Esto afecta la precisión de la predicción.

[0144] Como se usa en esta divulgación, las “muestras de borde de un bloque” generalmente se refieren a las muestras en el bloque que están adyacentes a uno de los cuatro límites del bloque, tales como las muestras en la primera y última fila del bloque y las muestras en la primera y última columna del bloque. Como se usa en esta divulgación, las muestras de borde superior, izquierdo, inferior y derecho de un bloque generalmente se refieren a las muestras en el bloque que están adyacentes a los límites superior, izquierdo, inferior y derecho del bloque, respectivamente. Tener en cuenta que la muestra de la esquina superior izquierda del bloque puede considerarse tanto como muestra de borde superior como muestra de borde izquierdo. Debe entenderse que, en algunos ejemplos, la muestra de la esquina superior izquierda puede considerarse una

muestra de borde superior y no una muestra de borde izquierdo; mientras que en otros ejemplos, la muestra de la esquina superior izquierda puede considerarse una muestra de borde izquierdo y no una muestra de borde superior. Consideraciones similares pueden aplicarse a las muestras de esquina superior derecha, inferior derecha e inferior izquierda del bloque.

[0145] Como se usa en esta divulgación, una banda de borde de muestras de un bloque generalmente se refiere a las muestras en el bloque que están en la vecindad de cualquiera de los cuatro límites del bloque, por ejemplo, muestras en las primeras o últimas pocas filas del bloque o las primeras o últimas pocas columnas del bloque. También se pueden definir definiciones similares para la banda del borde superior, izquierdo, derecho e inferior de muestras de un bloque. Como se usa en esta divulgación, una banda de borde n-superior de muestras de un bloque generalmente se refiere a las muestras que pertenecen a las n filas superiores del bloque, y una banda de borde n inferior de muestras de un bloque se define como las muestras que pertenecen a las n filas inferiores del bloque. Una banda de borde n izquierdo de muestras de un bloque se define como las muestras que pertenecen a las n columnas izquierdas del bloque, y una banda del borde n derecho de muestras de un bloque se define como las muestras que pertenecen a las n columnas derechas del bloque. En estos ejemplos, n será un número entero.

[0146] La FIG. 15 ilustra algunos ejemplos de las bandas de límite definidas anteriormente. Por ejemplo, el bloque 1502 (que se muestra con la línea negra en negrita) incluye muestras de borde izquierdo 1504, que se muestran en gris. El bloque 1506 (que se muestra con la línea negra en negrita) incluye la banda de 3 bordes superiores de las muestras 1512 y la banda de 2 bordes inferiores de las muestras 1514.

[0147] Esta divulgación describe técnicas que pueden mejorar la eficacia de ALWIP. Las siguientes técnicas descritas se pueden usar por separado o en combinación.

[0148] En algún ejemplo, el proceso de sobremuestreo descrito anteriormente puede modificarse de modo que se pueda reducir el error de predicción de las muestras. Por ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar una etapa de filtrado adicional para reducir el error de predicción. En algunos ejemplos, esta etapa de filtrado adicional puede incorporarse efectivamente en la etapa de interpolación o sobremuestreo, de modo que el filtrado adicional es parte de la etapa de interpolación o sobremuestreo, en lugar de estar separada de ella. En otros ejemplos, el filtrado adicional puede realizarse en lugar de la etapa de interpolación o sobremuestreo.

[0149] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar, después de aplicar la interpolación lineal en una o ambas direcciones para generar las muestras de predicción, un filtrado adicional en las muestras de borde superior e izquierdo. Por ejemplo, se puede aplicar un filtro $F1$ en las muestras de borde superior en la dirección vertical y se puede aplicar un filtro $F2$ en las muestras de borde izquierdo en la dirección horizontal. El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar todas las muestras de referencia de límite en la etapa de filtro adicional. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden configurarse para aplicar un filtro de submuestreo diferente a las muestras de límite para generar un conjunto de muestras de referencia de límite reducido para la etapa de filtro adicional.

[0150] El codificador de video 200 y el decodificador de video 300 pueden configurarse para seleccionar los filtros $F1$ y $F2$ de un conjunto de filtros que pueden señalizarse o predeterminarse. Un conjunto no exhaustivo de coeficientes de tales filtros es el siguiente:

1. [1 2 1]
2. [1]
3. [1 2 2 2 1]
4. [1 4 6 4 1]
5. [1 3]

[0151] En el ejemplo anterior, un filtro de [1] puede ser efectivamente un filtro de “copia” que copia, sin promediar, un valor de muestra al que se aplica el filtro. Por el contrario, un filtro [1 1 1 1] puede representar un filtrado promedio. Los filtros con otros valores pueden representar filtros de promedio ponderado. Los filtros 1-4 anteriores son filtros simétricos, de modo que el coeficiente medio (por ejemplo, 2 en el filtro 1, 6 en el filtro 4, etc.) se aplica a la muestra que se filtra.

[0152] En algunos ejemplos, el filtro $F1$ y $F2$ pueden no ser el mismo. En otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden seleccionar el filtro en función de los factores de sobremuestreo utilizados en el proceso de sobremuestreo de ALWIP. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden aplicar la etapa de filtro adicional a las muestras de borde superior del bloque solo cuando el factor de sobremuestreo es más de 1 en la dirección horizontal. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden aplicar la etapa de filtro adicional a las muestras de borde izquierdo del bloque solo cuando el factor de sobremuestreo es más de 1 en la dirección vertical.

[0153] En algunos ejemplos, la etapa de filtro adicional puede realizarse de manera similar a las operaciones de PDPC, donde las muestras predichas se actualizan con un promedio ponderado de la predicción y las muestras de límite.

[0154] La FIG. 16 muestra un ejemplo en el que las muestras de referencia de límite se usan sin submuestreo y hay un valor intermedio de muestras predichas en la parte inferior derecha. UpV y UpH indican los factores de sobremuestreo en las direcciones vertical y horizontal.

[0155] En este ejemplo, dejar que $\text{pred}(x,y)$ sea la predicción obtenida como resultado de la interpolación lineal ALWIP. La etapa adicional modifica el $\text{pred}(x,y)$ de la siguiente manera.

$$\text{Pred}(x,y) = (wT * BT(x) + wL * BL(y) + (64 - wT - wL) * \text{Pred}(x,y) + 32) \gg 6$$

[0156] Tener en cuenta que los valores de 32 y 64 anteriores elegidos se basan en la precisión de los valores wL y wT , y pueden ser diferentes para diferentes precisiones de wT y wL . En este ejemplo, se supone que los valores de wT y wL están en el rango de 0 a 64; en algunos casos, wT , wL y $64-wT-wL$ están restringidos para que sean no negativos. Los pesos se pueden derivar de la siguiente manera:

$$wT = 32 \gg ((y \ll 1) \gg \text{shift}), wL = 32 \gg ((x \ll 1) \gg \text{shift})$$

donde el valor de shift puede ser fijo o derivado usando el ancho y el alto del bloque; por ejemplo, como $\text{shift} = (\log_2(\text{width}) + \log_2(\text{height}) + 2) \gg 2$,

[0157] En algunos casos, el valor de las muestras predichas correspondientes a las posiciones predichas intermedias (por ejemplo, P en la FIG. 16) no se modifica.

[0158] Cuando las muestras de modificación están en otras partes del bloque, solo se pueden usar las muestras de límite superior o izquierdo para la etapa de filtro adicional. Por ejemplo, para una muestra (x,y) con respecto a la muestra superior izquierda del bloque, cuando el valor de x es mayor o igual que un valor umbral (por ejemplo, UpH), el valor de wL puede establecerse igual a 0; de manera similar cuando el valor de y es mayor o igual que un valor umbral (por ejemplo, UpV), el valor de wT puede establecerse igual a 0.

[0159] En algunos ejemplos, la etapa de filtro adicional solo se aplica para modificar el valor de la posición de las muestras en las muestras de borde n1 superior y muestras de borde n2 izquierdo del bloque, donde el valor de $n1$ y $n2$ puede determinarse mediante el factor de sobremuestreo para ALWIP en el bloque (por ejemplo, $n1$ puede ser igual a $\text{UpV}-1$ y $n2$ puede ser igual a $\text{UpH}-1$, donde UpV y UpH son los factores de sobremuestreo en las direcciones vertical y horizontal, respectivamente).

[0160] En algunos ejemplos, el proceso de sobremuestreo se modifica de modo que las

muestras se predigan usando pesos dependientes de la posición, o en otras palabras, la interpolación lineal y la operación de etapa adicional se combinan en un solo paso.

[0161] La predicción de todas las muestras en el bloque podría generalizarse de la siguiente manera:

Cuando (x,y) no pertenece a las muestras predichas intermedias, el valor $\text{pred}(x,y)$ se determina de la siguiente manera (x e y son con respecto a la muestra en la parte inferior derecha de $P3$):

$$\text{Pred}(x,y) = (w1 * P1 + w2 * P2 + w3 * P3 + w4 * P4 + wL * L + wT * T + \text{offset})$$

>> shift

donde el valor de desplazamiento y shift se elige para normalizar los valores de las muestras predichas y los valores se establecen de la siguiente manera: $w1 = x*y$, $w2 = (UpH-1-x)*y$, $w3 = (UpH-1-x)*(UpV-1-y)$, $w4 = x*(UpH-1-y)$ y wL y wT se determinan con base en ecuaciones similares a PDPC, con las siguientes excepciones:

- Cuando L y T pertenecen a una referencia de límite, $w2$, $w3$ y $w4$ se establecen igual a 0.
- De lo contrario, si L no pertenece a un límite y T pertenece a un límite, wL , $w3$ y $w4$ se establecen igual a 0.
- De lo contrario, si L pertenece a un límite y T no pertenece a un límite, wT , $w3$ y $w2$ se establecen igual a 0.
- De lo contrario, wL y wT se establecen igual a 0.

[0162] En algunos ejemplos, el valor de las muestras predichas se deriva de forma similar a la derivación de la predicción plana; por ejemplo, en la FIG. 17, el valor de las muestras en x,y se deriva utilizando una derivación similar a la predicción plana: derivando una predicción horizontal de L (o $P2$ y $P3$), $P4$ y $P1$, y derivando una predicción vertical de T (o $P3$ y $P4$), $P1$ y $P2$.

[0163] Tener en cuenta que el valor de $w1$, $w2$, $w3$, $w4$, wL y wT solo se ilustran como ejemplos, y se pueden elegir otros valores de estos pesos.

[0164] En algunos ejemplos, cuando se usan pesos dependientes de la posición para el proceso de sobremuestreo modificado, la elección de los pesos puede basarse en el modo/matriz particular que se usa con ALWIP. En algunos casos, se puede usar una tabla de mapeo para interpretar un modo de intrapredicción que corresponde a una matriz particular. Los pesos dependientes de la posición se pueden elegir en función del modo de intrapredicción interpretada, y se puede usar una o más muestras de referencia de límite para calcular el valor predicho. En algunos casos, se puede usar un conjunto predeterminado de pesos para los pesos dependientes de la posición,

independientemente de la matriz que se usa. En algunos ejemplos, los pesos dependientes de la posición también pueden depender de otras características que incluyen, pero sin limitación, la forma del bloque (ancho, alto), relación de aspecto, etc.

[0165] La FIG. 18 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede realizar las técnicas de esta divulgación. La FIG. 18 se proporciona con fines de explicación y no debe considerarse que limita las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 en el contexto de estándares de codificación de video tales como el estándar de codificación de video HEVC y el estándar de codificación de video H.266 en desarrollo. Sin embargo, las técnicas de esta divulgación no se limitan a estos estándares de codificación de video, y son aplicables generalmente a la codificación y decodificación de video.

[0166] En el ejemplo de la FIG. 18, el codificador de video 200 incluye memoria de datos de video 230, unidad de selección de modo 202, unidad de generación residual 204, unidad de procesamiento de transformada 206, unidad de cuantificación 208, unidad de cuantificación inversa 210, unidad de procesamiento de transformada inversa 212, unidad de reconstrucción 214, unidad de filtro 216, memoria intermedia de imagen decodificada (DPB) 218 y unidad de codificación por entropía 220.

[0167] La memoria de datos de video 230 puede almacenar datos de video para ser codificados por los componentes del codificador de video 200. El codificador de video 200 puede recibir los datos de video almacenados en la memoria de datos de video 230 de, por ejemplo, la fuente de video 104 (FIG. 1). 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 el codificador de video 200. La memoria de datos de video 230 y 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 DPB 218 pueden ser proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de video 230 puede estar en chip con otros componentes del codificador de video 200, como se ilustra, o fuera del chip en relación con esos componentes.

[0168] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describa específicamente como tal, o a la memoria externa al codificador de

video 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como memoria de referencia que almacena datos de video que el codificador de video 200 recibe para codificar (por ejemplo, datos de video para un bloque actual que se codificará). La memoria 106 de la FIG. 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

[0169] Las diversas unidades de la FIG. 18 se ilustran para ayudar a comprender las operaciones realizadas por el codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de los mismos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar software o firmware que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del software o firmware. Los circuitos de función fija pueden ejecutar instrucciones de software (por ejemplo, para recibir parámetros o emitir parámetros), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, la una o más unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la una o más unidades pueden ser circuitos integrados.

[0170] El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU), unidades de funciones elementales (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En los ejemplos en los que las operaciones del codificador de video 200 se realizan utilizando un software ejecutado por los circuitos programables, la memoria 106 (FIG. 1) puede almacenar el código objeto del software que el codificador de video 200 recibe y ejecuta, u otra memoria dentro del codificador de video 200 (no mostrado) puede almacenar tales instrucciones.

[0171] 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.

[0172] La unidad de selección de modo 202 incluye una unidad de estimación de movimiento 222, una unidad de compensación de movimiento 224 y una unidad de intrapredicción 226. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para realizar predicciones de video de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM), o similar.

[0173] 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 transformada 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.

[0174] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 230 en una serie de CTU, y encapsular una o más CTU dentro de un segmento. La unidad de selección de modo 202 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura QTBT o la estructura quadtree de HEVC descrita anteriormente. Como se describió anteriormente, el codificador de video 200 puede formar una o más CU a partir de la partición de una CTU de acuerdo con la estructura de árbol. Tal CU también se puede denominar generalmente “bloque de video” o “bloque”.

[0175] 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 cuadradas (SSD), la diferencia absoluta media (MAD), diferencias cuadradas medias (MSD) o similares. La unidad de estimación de movimiento 222 generalmente puede realizar estos cálculos usando diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se está considerando. La unidad de estimación de movimiento 222 puede identificar un bloque de referencia que tiene un valor más bajo que resulta de estos cálculos, lo que indica un bloque de referencia que coincide más estrechamente con el bloque actual.

[0176] 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 compensación de movimiento 224 puede generar entonces un bloque de predicción usando los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia usando el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por los respectivos vectores de movimiento y combinar los datos recuperados, por ejemplo, a través del promedio de muestra por muestra o el promedio ponderado.

[0177] 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. Como otro ejemplo, para un

modo ALWIP, la unidad de intrapredicción 226 puede derivar, con base en un conjunto de muestras vecinas de borde izquierdo del bloque actual y un conjunto de muestras vecinas de borde superior del bloque actual, un subconjunto de muestras de borde izquierdo y un subconjunto de muestras de borde superior; aplicar un modelo afín al subconjunto de muestras de borde izquierdo y al subconjunto de muestras de borde superior para generar un bloque intermedio de muestras intermedias; y filtrar, usando el conjunto de muestras vecinas de borde izquierdo y el conjunto de muestras vecinas de borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final.

[0178] La unidad de selección de modo 202 proporciona el bloque de predicción a la unidad de generación residual 204. La unidad de generación residual 204 recibe una versión sin codificar del bloque actual desde la memoria de datos de video 230 y el bloque de predicción desde la unidad de selección de modo 202. 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 usando uno o más circuitos sustractores que realizan sustracción binaria.

[0179] 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 tengan varios tamaños. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación luma de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es de $2N \times 2N$, 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.

[0180] En los ejemplos en los que la unidad de selección de modo no divide más una CU en PU, cada CU puede estar asociada con un bloque de codificación luma y con

bloques de codificación croma correspondientes. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación luma de la CU. El codificador de video 200 y el decodificador de video 300 pueden admitir tamaños CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

[0181] 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 en 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 con base en una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 a codificar.

[0182] Como se describió anteriormente, la unidad de generación residual 204 recibe los datos de video para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 genera 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.

[0183] La unidad de procesamiento de transformada 206 aplica una o más transformadas al bloque residual para generar un bloque de coeficientes de transformada (denominado en este documento “bloque de coeficiente de transformada”). La unidad de procesamiento de transformada 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformada. 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 transformadas a un bloque residual, por ejemplo, una transformada primaria y una transformada secundaria, tal como una transformada rotacional. En algunos ejemplos, la unidad de procesamiento de transformada 206 no aplica transformadas a un bloque residual.

[0184] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformada en un bloque de coeficiente de transformada, para producir un bloque de coeficiente de transformada cuantificado. La unidad de cuantificación 208 puede

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

[0185] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformada inversa 212 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficiente de transformada cuantificado, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficiente de transformada. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con 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.

[0186] La unidad de filtro 216 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de las CU. Las operaciones de la unidad de filtro 216 pueden omitirse, en algunos ejemplos.

[0187] 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.

[0188] En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía bloques de coeficientes de transformada cuantificados de 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 intramodal 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 operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de entropía de partición de intervalo de probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de codificación 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.

[0189] El codificador de video 200 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o una imagen. En particular, la unidad de codificación por entropía 220 puede emitir el flujo de bits.

[0190] Las operaciones descritas anteriormente 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 anteriormente, en algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una CU. En algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una PU.

[0191] 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.

[0192] El codificador de video 200 representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para realizar las técnicas de esta divulgación, incluidas las técnicas para sobremuestreo en intrapredicción ponderada lineal afín.

[0193] La FIG. 19 es un diagrama de bloques que ilustra un decodificador de video 300 de ejemplo que puede realizar las técnicas de esta divulgación. La FIG. 19 se proporciona con fines de explicación y no limita las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador de video 300 de acuerdo con las técnicas de JEM y HEVC. Sin embargo, las técnicas de esta divulgación pueden realizarse mediante dispositivos de codificación de video que están configurados para otros estándares de codificación de video.

[0194] En el ejemplo de la FIG. 19, 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. La unidad de procesamiento de predicción 304 incluye la unidad de compensación de movimiento 316 y la unidad de intrapredicción 318. La unidad de procesamiento de predicción 304 puede incluir unidades de adición para realizar la predicción de acuerdo con otros modos de predicción. Como ejemplos, la unidad de procesamiento de predicción 304 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede formar parte de la unidad de compensación de movimiento 316), una unidad afín, una unidad de modelo lineal (LM) o similares. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o diferentes componentes funcionales.

[0195] 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 de 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 la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de 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 en chip con otros componentes del decodificador de video 300, o fuera de chip en relación con esos componentes.

[0196] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (FIG. 1). Es decir, la memoria 120 puede almacenar datos como se discutió anteriormente 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 el software para ser ejecutado por circuitos de procesamiento del decodificador de video 300.

[0197] Las diversas unidades mostradas en la FIG. 19 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. 18, 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 emitir parámetros), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos,

la una o más de las unidades pueden ser bloques de circuito distintos (función fija o programable), y en algunos ejemplos, la una o más unidades pueden ser circuitos integrados.

[0198] El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables. En los ejemplos en los que las operaciones del decodificador de video 300 se realizan mediante software que se ejecuta en los circuitos programables, la memoria en chip o fuera de chip puede almacenar instrucciones (por ejemplo, código de objeto) del software que el decodificador de video 300 recibe y ejecuta.

[0199] La unidad de decodificación por entropía 302 puede recibir datos de video codificados de 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 basados en los elementos de sintaxis extraídos del flujo de bits.

[0200] 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, decodificando, puede denominarse un “bloque actual”).

[0201] La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformada cuantificados de un bloque de coeficiente de transformada cuantificado, así como información de transformada, tal como un parámetro de cuantificación (QP) y/o indicaciones de modo de transformada. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficiente de transformada cuantificado para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para aplicar la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda bit a bit para cuantificar inversamente los coeficientes de transformada cuantificados. La unidad de cuantificación inversa 306 puede formar así un bloque de coeficiente de transformada que incluye coeficientes de transformada.

[0202] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente de transformada, la unidad de procesamiento de transformada inversa 308 puede aplicar una o más transformadas inversas al bloque de coeficiente de transformada para generar un bloque residual asociado con el bloque actual. Por

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

[0203] 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 la DPB 314 desde la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede realizar el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de compensación de movimiento 224 (FIG. 18).

[0204] 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. Como otro ejemplo, si los elementos de sintaxis de la información de predicción indican que el bloque actual es intrapredicho en un modo ALWIP, entonces la unidad de intrapredicción 318 puede derivar, con base en un conjunto de muestras vecinas de borde izquierdo del bloque actual y un conjunto de muestras vecinas de borde superior del bloque actual, un subconjunto de muestras de borde izquierdo y un subconjunto de muestras de borde superior; aplicar un modelo afín al subconjunto de muestras de borde izquierdo y al subconjunto de muestras de borde superior para generar un bloque intermedio de muestras intermedias; y filtrar, usando el conjunto de muestras vecinas de borde izquierdo y el conjunto de muestras vecinas de borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final. 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. 18). La unidad de interpredicción 318 puede recuperar datos de muestras vecinas al bloque actual de la DPB 314.

[0205] La unidad de reconstrucción 310 puede reconstruir el bloque actual usando el bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0206] La unidad de filtro 312 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de los bloques reconstruidos. Las operaciones de la unidad de filtro 312 no se realizan necesariamente en todos los ejemplos.

[0207] El decodificador de video 300 puede almacenar los bloques reconstruidos en la DPB 314. Como se discutió anteriormente, la DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes decodificadas previamente para compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas desde la DPB para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la FIG. 1.

[0208] De esta manera, el decodificador de video 300 representa un ejemplo de un dispositivo de decodificación de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para realizar las técnicas de esta divulgación, incluidas las técnicas para sobremuestreo en intrapredicción ponderada lineal afín.

[0209] La FIG. 20 es un diagrama de flujo que ilustra un proceso de ejemplo para codificar un bloque actual. El bloque actual puede incluir una CU actual. Aunque se describe con respecto al codificador de video 200 (FIG. 1 y 18), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la FIG. 20.

[0210] En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual (350). Por ejemplo, el codificador de video 200 puede formar un bloque de predicción para el bloque actual usando cualquiera de las técnicas de intrapredicción descritas en esta divulgación. El codificador de video 200 puede calcular entonces un bloque residual para el bloque actual (352). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque 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 (354). A continuación, el codificador de video 200 puede explorar los coeficientes de transformada cuantificados del bloque residual (356). Durante la exploración, o después de la exploración, el codificador de

video 200 puede codificar por entropía los coeficientes (358). Por ejemplo, el codificador de video 200 puede codificar los coeficientes usando CAVLC o CABAC. El codificador de video 200 puede entonces emitir los datos codificados por entropía del bloque (360).

[0211] La FIG. 21 es un diagrama de flujo que ilustra un proceso de ejemplo para decodificar un bloque actual de datos de video. El bloque actual puede incluir una CU actual. Aunque se describe con respecto al decodificador de video 300 (FIG. 1 y 19), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la FIG. 21.

[0212] El decodificador de video 300 puede recibir datos codificados por entropía para el bloque actual, tales como información de predicción codificada por entropía y datos codificados por entropía para coeficientes de un bloque residual correspondiente al bloque actual (370). El decodificador de video 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y para reproducir los coeficientes del bloque residual (372). El decodificador de video 300 puede predecir el bloque actual (374), por ejemplo, usando un modo de 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 predecir, por ejemplo, el bloque actual usando cualquiera de las técnicas de intrapredicción descritas en esta divulgación. El decodificador de video 300 puede entonces explorar inversamente los coeficientes reproducidos (376), para crear un bloque de coeficientes de transformada cuantificados. El decodificador de video 300 puede entonces cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (378). El decodificador de video 300 puede decodificar finalmente el bloque actual combinando el bloque de predicción y el bloque residual (380).

[0213] La FIG. 22 es un diagrama de flujo que ilustra un proceso de ejemplo para decodificar un bloque actual de datos de video. El bloque actual puede incluir una CU actual. Aunque se describe con respecto al decodificador de video 300 (FIG. 1 y 19), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la FIG. 22. Por ejemplo, el bucle de decodificación del codificador de video 200 (FIG. 1 y 18) también puede realizar las técnicas de la FIG. 22.

[0214] En el ejemplo de la FIG. 22, el decodificador de video 300 determina que un bloque actual de datos de video está codificado en un modo ALWIP (382). El decodificador de video 300 deriva, con base en un conjunto de muestras vecinas de borde izquierdo del bloque actual y un conjunto de muestras vecinas de borde superior

del bloque actual, un subconjunto de muestras de borde izquierdo y un subconjunto de muestras de borde superior (384). El decodificador de video 300 puede derivar, por ejemplo, el subconjunto de muestras de borde izquierdo mediante submuestreo, usando el promedio, del conjunto de muestras vecinas de borde izquierdo y derivar el subconjunto de muestras de borde superior mediante submuestreo, usando el promedio, del conjunto de muestras vecinas de borde superior.

[0215] El conjunto de muestras vecinas de borde superior puede tener, por ejemplo, N muestras totales y el subconjunto de muestras de borde superior puede tener un total de $N/2$ muestras, donde N es un número entero que representa un número de columnas incluidas en el bloque actual. El conjunto de muestras vecinas del borde izquierdo puede tener, por ejemplo, N muestras totales y el subconjunto de muestras de borde izquierdo puede tener un total de $N/2$ muestras, donde N es un número entero que representa un número de filas incluidas en el bloque actual. En un ejemplo, el bloque actual puede ser un bloque de muestras de $N \times M$, donde N es un valor entero que representa un número de columnas en el bloque actual y un número de muestras en el conjunto de muestras de borde superior y M es un valor entero que representa un número de filas en el bloque actual y un número de muestras en el conjunto de muestras de borde izquierdo. N y M pueden o no ser iguales. El conjunto de muestras de borde izquierdo puede tener $M/2$ muestras, y el conjunto de muestras de borde superior puede tener $N/2$ muestras.

[0216] El decodificador de video 300 aplica un modelo afín al subconjunto de muestras de borde izquierdo y al subconjunto de muestras de borde superior para generar un bloque intermedio de muestras intermedias (386). Para aplicar el modelo afín al subconjunto de muestras de borde izquierdo y al subconjunto de muestras de borde superior para generar el bloque intermedio de muestras intermedias, el decodificador de video 300 puede multiplicar, por ejemplo, el subconjunto de muestras de borde izquierdo y el subconjunto de muestras de borde superior por una matriz y un vector de sesgo.

[0217] El decodificador de video 300 filtra, usando el conjunto de muestras vecinas de borde izquierdo y el conjunto de muestras vecinas de borde superior del bloque actual, las muestras intermedias para generar un bloque de predicción final (388). Para filtrar las muestras intermedias para generar el bloque de predicción final, el decodificador de video 300 puede sobremuestrear, por ejemplo, las muestras intermedias utilizando un segundo subconjunto de muestras de borde izquierdo que es diferente al subconjunto de muestras de borde izquierdo y un segundo subconjunto de muestras de borde superior que es diferente al subconjunto de muestras de borde superior. El decodificador de video 300 puede sobremuestrear, por ejemplo, las muestras intermedias según los

valores de muestra reales en el conjunto de muestras de borde superior o el conjunto de muestras de borde izquierdo, en oposición a los valores de muestra del subconjunto de muestras de borde superior o subconjunto de muestras de borde izquierdo, donde los subconjuntos se obtienen promediando y son diferentes a los conjuntos. Para filtrar las muestras intermedias para generar un bloque de predicción final, el decodificador de video 300 puede aplicar uno o más filtros en una dirección vertical y uno o más filtros en una dirección horizontal. Para aplicar el uno o más filtros en la dirección vertical, el decodificador de video 300 puede usar muestras del conjunto de muestras de borde superior para realizar una interpolación lineal en la dirección vertical. Es decir, si el subconjunto de muestras de borde superior incluye $M/2$ muestras, entonces el decodificador de video 300 puede usar un subconjunto diferente de las M muestras de borde superior cuando aplica el uno o más filtros en la dirección vertical. Las muestras utilizadas para la interpolación pueden ser muestras reales de las M muestras de borde superior en lugar de muestras determinadas a partir del promedio. Como parte de la aplicación del uno o más filtros en la dirección vertical, el decodificador de video 300 puede comprender aplicar pesos dependientes de la posición a al menos algunas de las M muestras del conjunto de muestras de borde superior.

[0218] El decodificador de video 300 decodifica el bloque actual de datos de video basado en el bloque de predicción final (390). Para decodificar el bloque actual de datos de video basado en el bloque de predicción final, el decodificador de video 300 puede determinar valores residuales para el bloque actual de datos de video; agregar los valores residuales al bloque de predicción filtrado para determinar un bloque reconstruido para el bloque actual de datos de video; y aplicar uno o más filtros al bloque reconstruido para generar un bloque decodificado de datos de video. El decodificador de video 300 puede entonces emitir, para visualización y/o almacenamiento, una imagen que incluye el bloque decodificado de datos de video. El decodificador de video 300 puede almacenar, por ejemplo, una copia de la imagen para usar en la decodificación de otras imágenes de los datos de video.

[0219] Debe reconocerse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través del procesamiento de subprocesos múltiples, procesamiento de interrupción o procesadores múltiples, en lugar de secuencialmente.

[0220] 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 informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios tangibles de almacenamiento legibles por computadora que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder una o más computadoras o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por computadora.

[0221] A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por computadora pueden incluir uno o más de RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que puede usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que puede acceder una computadora. Además, cualquier conexión se denomina correctamente un medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota utilizando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, 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 (en inglés, disk) y disco (en inglés, disc), como se usan en este documento, incluyen disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos

(en inglés, disks) generalmente reproducen datos magnéticamente, mientras que los discos (en inglés, discs) reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computadora.

[0222] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más DSP, microprocesadores de propósito general, ASIC, FPGA u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término “procesador”, como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos dedicados de hardware y/o software configurados para codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

[0223] Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Por el contrario, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describe anteriormente, junto con el software y/o firmware adecuado.

[0224] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.