

**MÉTODO, DISPOSITIVO Y MEDIO DE ALMACENAMIENTO LEGIBLE
POR COMPUTADORA PARA LA INTRAPREDICCIÓN PONDERADA
LINEAL AFÍN EN CODIFICACIÓN DE VIDEO**

[0001] La presente solicitud reivindica la prioridad de la Solicitud de Patente US N.º 16/867.208, presentada el 5 de mayo de 2020, que reivindica el beneficio de la Solicitud de Patente Provisional US N.º 62/845.790, presentada el 9 de mayo de 2019, y la Solicitud de Patente Provisional US N.º 62/864.320, presentada el 20 de junio de 2019, cuyo contenido completo se incorpora por referencia.

CAMPO TÉCNICO

[0002] La presente divulgación se refiere al encriptado de video y decodificación de video.

ANTECEDENTES

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

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

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

SÍNTESIS

[0005] En general, esta divulgación describe técnicas para la intrapredicción, que incluyen la derivación y la señalización de modos para la intrapredicción ponderada lineal. Por ejemplo, en un ejemplo, esta divulgación describe un codificador de video, como un codificador de video o un decodificador de video, que realiza un proceso de derivación del modo más probable (MPM) que deriva uno o más MPM para un bloque actual que no está codificado con el uso de la intrapredicción ponderada lineal afín (ALWIP). Los MPM para el bloque actual pueden ser modos de intrapredicción determinados para ser usados con mayor probabilidad para codificar el bloque actual. Como parte de la realización del proceso de derivación de MPM, el codificador de video determina si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. En función de que el bloque vecino es un bloque vecino codificado con ALWIP, el codificador de video determina que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. El codificador de video codifica el bloque actual en función de uno de los MPM para el bloque actual.

[0006] En un ejemplo, esta divulgación describe un método de codificación de datos de video, en donde el método comprende: realizar un proceso de derivación del modo más probable (MPM) que deriva uno o más MPM para un bloque actual de los datos de video, en donde el bloque actual no se codifica con el uso de la intrapredicción ponderada lineal afín (ALWIP), y realizar el proceso de derivación de MPM comprende: determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP; y en función de que el bloque vecino es un bloque vecino codificado con ALWIP: determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado; e incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM; y codificar el bloque actual en función de uno de los MPM para el bloque actual.

[0007] En otro ejemplo, esta divulgación describe un dispositivo para codificar datos de video, en donde el dispositivo comprende: una memoria para almacenar los datos de

video; y uno o más procesadores implementados en circuitos, el procesador o los procesadores configurados para: realizar un proceso de derivación del modo más probable (MPM) que deriva uno o más MPM para un bloque actual de datos de video, en donde el bloque actual no está codificado con el uso de la intrapredicción ponderada lineal afín (ALWIP) y realizar el proceso de derivación de MPM comprende: determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP; y en función de que el bloque vecino es un bloque vecino codificado con ALWIP: determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado; e incluir un modo de intrapredicción correspondiente a los valores predeterminados como uno del MPM o los MPM; y codificar el bloque actual en función de uno de los MPM para el bloque actual.

[0008] En otro ejemplo, esta divulgación describe un dispositivo para codificar datos de video, en donde el dispositivo comprende: medios para realizar un proceso de derivación del modo más probable (MPM) que deriva uno o más MPM para un bloque actual de datos de video, en donde el bloque actual no se codifica con el uso de la intrapredicción ponderada lineal afín (ALWIP), y los medios para realizar el proceso de derivación de MPM comprenden: medios para determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP; y medios para determinar, en función de que el bloque vecino es un bloque vecino codificado con ALWIP, que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado e incluye un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM; y medios para codificar el bloque actual en función de uno de los MPM para el bloque actual.

[0009] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por computadora que tiene almacenadas instrucciones que, cuando se ejecutan, hacen que uno o más procesadores: realicen un proceso de derivación del modo más probable (MPM) que deriva uno o más MPM para un bloque actual de datos de video, en donde el bloque actual no se codifica con el uso de la intrapredicción ponderada lineal afín (ALWIP), y la realización del proceso de derivación MPM comprende: determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP; y en función de que el bloque vecino es un bloque vecino codificado con ALWIP: determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado; e incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM; y codificar el bloque actual en función de uno de los MPM para el bloque actual.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

[0012] La Figura 2 es un diagrama de bloques que ilustra un ejemplo de intrapredicción, con flechas que apuntan hacia muestras de referencia.

[0013] La Figura 3 es un diagrama conceptual de un ejemplo de bloque rectangular de 8x4 donde no se usan muestras de referencia “más cercanas”, pero se pueden usar muestras de referencia más lejanas debido a una restricción de una dirección de intrapredicción para que esté en el intervalo de -135 grados a 45 grados.

[0014] La Figura 4 es un diagrama conceptual que ilustra ángulos amplios (-1 a -10 y 67 a 76) representados además de los 65 modos angulares.

[0015] Las Figuras 5A, 5B y 5C son ilustraciones de procesos de mapeo de modo para modos de intrapredicción fuera de un intervalo de dirección diagonal.

[0016] La Figura 6 es un diagrama conceptual que ilustra ángulos amplios (-1 a -14 y 67 a 80) en el modelo de prueba 3 de VVC (VTM3) más allá de los modos 2 y 66 para un total de 93 modos angulares.

[0017] La Figura 7 es una tabla que proporciona una especificación de ángulos de intrapredicción.

[0018] La Figura 8 es una ilustración de muestras de referencia de múltiples líneas de referencia que pueden usarse para la intrapredicción del bloque de codificación.

[0019] La Figura 9 es un diagrama conceptual que ilustra un ejemplo de división de bloques de 4x8 y 8x4.

[0020] La Figura 10 es un diagrama conceptual que ilustra un ejemplo de división de todos los bloques excepto los de 4x8, 8x4 y 4x4.

[0021] La Figura 11 es un diagrama conceptual que ilustra un ejemplo de un proceso de intrapredicción ponderada lineal afín (ALWIP) en un bloque de 8x8.

[0022] La Figura 12 es una tabla que ilustra el número de parámetros de matriz y parámetros de desviación para cada tipo de ALWIP en el Borrador de trabajo de codificación de video versátil 4 (VVC WD4).

[0023] La Figura 13 es un diagrama conceptual que ilustra bloques vecinos de ejemplo utilizados para derivar los modos más probables (MPM) de ALWIP para un bloque

actual, de acuerdo con una o más técnicas de esta divulgación.

[0024] Las Figuras 14A y 14B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario quadtree (QTBT) y una unidad de árbol de codificación (CTU) correspondiente.

[0025] La Figura 15 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[0026] La Figura 16 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[0027] La Figura 17 es un diagrama de flujo que ilustra un método de ejemplo para encriptar un bloque actual.

[0028] La Figura 18 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de video.

[0029] La Figura 19 es un diagrama de flujo que ilustra un método de ejemplo de codificación de datos de video, de acuerdo con una o más técnicas de esta divulgación.

[0030] La Figura 20 es un diagrama de flujo que ilustra un método de ejemplo de codificación de un bloque de croma, de acuerdo con una o más técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0031] La intrapredicción ponderada lineal afín (ALWIP) es una forma de intrapredicción en la que se genera un bloque de predicción mediante el muestreo descendente de muestras de referencia de un bloque, la multiplicación de un vector de las muestras de referencia muestreadas de manera descendente por una matriz y la adición de un vector de sesgo, el mapeo de los valores de muestra resultantes en posiciones predeterminadas dentro del bloque de predicción y la realización de una interpolación lineal en los valores de muestra en el bloque de predicción para determinar valores de muestra para las posiciones restantes dentro del bloque de predicción. El uso de ALWIP puede conducir a mejores eficiencias de codificación para algunos tipos de bloques, pero las técnicas de intrapredicción regulares pueden conducir a mejores eficiencias de codificación que ALWIP para otros tipos de bloques. Por lo tanto, algunos bloques dentro de una imagen pueden codificarse con el uso de ALWIP y algunos bloques dentro de una imagen pueden codificarse con el uso de técnicas de intrapredicción regulares, como un modo de intrapredicción planar, un modo de intrapredicción DC o un modo de intrapredicción direccional. Aunque el ejemplo anterior describe varios pasos para ALWIP, algunas implementaciones pueden optar por excluir uno o más de los pasos anteriores, y otras implementaciones pueden incluir otros pasos para el proceso de

derivación. Sin embargo, una idea central en ALWIP es multiplicar un vector de muestras derivadas de muestras de referencia con una matriz (u operación equivalente).

[0032] Para mejorar la eficiencia de la señalización de un modo de intrapredicción (por ejemplo, un modo de intrapredicción planar, DC o direccional), un codificador de video (por ejemplo, un codificador de video o un decodificador de video) puede realizar un proceso de derivación de modo más probable (MPM) que deriva uno o más MPM para un bloque actual de datos de video. Cada uno de los MPM puede ser un modo de intrapredicción. Si el modo de intrapredicción óptimo para el bloque actual es uno de los MPM, se puede señalar un índice que indique uno de los MPM en un flujo de bits. Si el modo de intrapredicción óptimo para el bloque actual no es uno de los MPM, se puede señalar un índice que indique uno de los modos de intrapredicción que no son MPM. Debido a que hay menos MPM que modos de intrapredicción que no son MPM, un índice que indica un MPM puede codificarse con el uso de significativamente menos bits que un índice que indica uno de los modos de intrapredicción que no son MPM. Por lo tanto, el uso del proceso de derivación MPM puede incrementar la eficiencia de codificación para algunos bloques.

[0033] El codificador de video puede derivar los MPM en función de los modos de intrapredicción de bloques vecinos al bloque actual. Por ejemplo, si un bloque vecino izquierdo se codifica con el uso de un modo de intrapredicción DC, el codificador de video puede determinar que uno de los MPM es el modo de intrapredicción DC. Sin embargo, en el proceso de derivación de MPM, si uno de los bloques vecinos se codifica con el uso de ALWIP, el codificador de video realiza un proceso que usa datos de mapeo que mapea los parámetros ALWIP del bloque vecino a uno de los modos de intrapredicción direccional. Los parámetros ALWIP incluyen coeficientes de matriz y un vector de sesgo. El codificador de video puede entonces usar el modo de intrapredicción direccional determinado como uno de los MPM.

[0034] El almacenamiento de los datos de mapeo aumenta los requisitos de almacenamiento de datos de los encriptadores y decodificadores de video. Los mayores requisitos de almacenamiento de datos pueden incrementar el costo y la complejidad de los encriptadores y decodificadores de video. Además, las operaciones de búsqueda que se realizan para utilizar los datos de mapeo llevan tiempo, lo que puede ralentizar los procesos de encriptado y decodificación. Además, un bloque de predicción generado con el uso de un modo de intrapredicción determinado con el uso de los datos de mapeo es solo una aproximación del bloque de predicción que se generaría si se usaran los parámetros ALWIP del bloque vecino. Esta aproximación puede disminuir la eficiencia

de la codificación.

[0035] Las técnicas de la presente divulgación pueden abordar uno o más de estos problemas. Por ejemplo, como se describe en un ejemplo de esta divulgación, un codificador de video (por ejemplo, un codificador de video o un decodificador de video) puede realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de datos de video. En este ejemplo, el bloque actual no se codifica mediante ALWIP. Como parte de la realización del proceso de derivación de MPM, el codificador de video puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. En función de que el bloque vecino es un bloque vecino codificado con ALWIP, el codificador de video puede determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. Por consiguiente, el codificador de video puede incluir un modo de intrapredicción correspondiente al valor predeterminado como uno de los MPM. El codificador de video puede codificar el bloque actual en función de uno de los MPM para el bloque actual.

[0036] Al usar el modo de intrapredicción correspondiente al valor predeterminado como uno de los MPM cuando el bloque vecino está codificado con ALWIP, el codificador de video puede evitar la necesidad de almacenar los datos de mapeo. Evitar la necesidad de almacenar los datos de mapeo puede reducir los requisitos de almacenamiento de datos del codificador de video, lo que puede reducir el costo y la complejidad del codificador de video. Además, evitar el uso de los datos de mapeo puede acelerar los procesos de encriptado y decodificación al evitar las operaciones de búsqueda involucradas con el uso de los datos de mapeo.

[0037] La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de encriptado y decodificación de video 100 que puede realizar las técnicas de la presente divulgación. Las técnicas de la presente divulgación están generalmente dirigidas a la codificación (encriptado 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 y sin procesar, video codificado, video decodificado (por ejemplo, reconstruido) y metadatos de video, como datos de señalización.

[0038] Como se muestra en la Figura 1, el sistema 100 incluye un dispositivo de origen 102 que proporciona datos de video codificados para ser decodificados y visualizados por un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo de origen 102 proporciona los datos de video al dispositivo de destino 116 a través de un medio legible por computadora 110. El dispositivo de origen 102 y el dispositivo de destino 116

pueden incluir cualquiera de una amplia gama de dispositivos, que incluye computadoras de escritorio, computadoras portátiles (es decir, computadoras portátiles), decodificadores, dispositivos móviles tales como microteléfonos (por ejemplo, teléfonos inteligentes) y tablets, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de video de flujo continuo, un dispositivo receptor de difusión, 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.

[0039] En el ejemplo de la Figura 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 realizar la intrapredicción, incluida la derivación y señalización de modos para la intrapredicción ponderada lineal. Por lo tanto, el dispositivo de origen 102 representa un ejemplo de un dispositivo de encriptado 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.

[0040] El sistema 100 como se muestra en la Figura 1 es simplemente un ejemplo. En general, cualquier dispositivo de encriptado y/o decodificación de video digital puede realizar técnicas para realizar intrapredicción, incluida la derivación y señalización de modos para intrapredicción ponderada lineal. El dispositivo de origen 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo de origen 102 genera datos de video codificados para su transmisión al dispositivo de destino 116. La presente divulgación se refiere a un dispositivo de "codificación" como un dispositivo que realiza la codificación (encriptado y/o decodificación) de datos. Por lo tanto, el codificador de video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un encriptador de video y un decodificador de video, respectivamente. En algunos

ejemplos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden funcionar de una manera sustancialmente simétrica de manera que tanto el dispositivo de origen 102 como el dispositivo de destino 116 incluyen componentes de encriptado y decodificación de video. Por lo tanto, el sistema 100 puede admitir 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 de flujo continuo, reproducción de video, transmisión de video o telefonía de video.

[0041] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin codificar y sin procesar) y proporciona una serie secuencial de imágenes (también denominadas “cuadros”) de los datos de video al codificador de video 200, que codifica datos para las imágenes. La fuente de video 104 del dispositivo de origen 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video sin procesar previamente capturado, y/o una interfaz de alimentación de video para recibir video de un proveedor de contenido de video. Como alternativa adicional, la fuente de video 104 puede generar datos basados en gráficos de computadora como el video fuente, o una combinación de video en vivo, video archivado y video generado por computadora. En cada caso, el codificador de video 200 codifica los datos de video capturados, capturados previamente o generados por computadora. El codificador de video 200 puede reorganizar las imágenes del orden recibido (a veces denominado “orden de visualización”) en un orden de codificación para la codificación. El codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados. El dispositivo de origen 102 puede entonces emitir los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computadora 110 para recepción y/o recuperación por, por ejemplo, la interfaz de entrada 122 del dispositivo de destino 116.

[0042] La memoria 106 del dispositivo de origen 102 y la memoria 120 del dispositivo de destino 116 representan memorias de uso general. En cierto ejemplo, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video sin procesar de la fuente de video 104 y datos de video decodificados sin procesar del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de *software* ejecutables, por ejemplo, por el codificador de video 200 y decodificador de video 300, respectivamente. Aunque 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 uno o más búferes de video, por ejemplo, para almacenar datos de video sin procesar, decodificados y/o codificados.

[0043] El medio legible por computadora 110 puede representar cualquier tipo de medio o dispositivo capaz de transportar los datos de video codificados desde el dispositivo de origen 102 al dispositivo de destino 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para habilitar el dispositivo de origen 102 para transmitir datos de video codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red informática. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video codificados, y la interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede incluir cualquier medio de comunicación inalámbrico o cableado, como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 102 al dispositivo de destino 116.

[0044] En algunos ejemplos, el medio legible por computadora 110 puede incluir el dispositivo de almacenamiento 112. El dispositivo de origen 102 puede emitir datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a los datos codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados.

[0045] En algunos ejemplos, el medio legible por computadora 110 puede incluir el 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 origen 102 puede enviar datos de video codificados al servidor de archivos

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

[0046] 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 incluyen 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 incluye 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.

[0047] Las técnicas de la presente divulgación se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de flujo continuo por Internet, tales como transmisión de flujo continuo dinámica adaptativa sobre HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

[0048] 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 un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánico (OLED) u otro tipo de dispositivo de visualización.

[0049] Aunque no se muestra en la Figura 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).

[0050] El codificador de video 200 y el decodificador de video 300 pueden implementarse como cualquiera de una variedad de circuitos de codificador y/o decodificador adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados específicos de aplicación (ASIC), matrices de puerta programables en campo (FPGA), lógica discreta, *software*, *hardware*, *firmware* o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en

el *software*, un dispositivo puede almacenar instrucciones para el *software* en un medio adecuado, no transitorio, legible por computadora y ejecutar las instrucciones en *hardware* con el uso de uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de video 200 como el decodificador de video 300 pueden incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de video 200 y/o un decodificador de video 300 puede incluir un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

[0051] El codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también conocido como Codificación de Video de Alta Eficiencia (HEVC) o extensiones de este, como las extensiones de codificación de vista múltiple y/o de video escalable. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con otros estándares patentados o industriales, como el Modelo de Prueba de Exploración Conjunta (JEM) o ITU-T H.266, también conocido como Codificación de Video Versátil (VVC). Un borrador del estándar VVC se describe en Bross, *et al.* “*Versatile Video Coding (Draft 4)*”, Equipo Conjunto de Expertos en Video (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13ª reunión: Marrakech, MA, 9-18 de enero de 2019, JVET-M1001 (en adelante, “VVC WD4”). Otro borrador del estándar VVC se describe en Bross, *et al.* “*Versatile Video Coding (Draft 5)*”, Equipo Conjunto de Expertos en Video (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 14ª reunión: Ginebra, CH, 19-27 de marzo de 2019, JVET-N1001-v5 (en adelante, “VVC Draft 5”). Un objetivo de VVC es proporcionar una mejora significativa en el rendimiento de compresión sobre el estándar HEVC existente, ayudando así en la implementación de servicios de video de mayor calidad y aplicaciones emergentes como multimedia inmersiva omnidireccional de 360 ° y video de alto rango dinámico (HDR). Sin embargo, las técnicas de la presente divulgación no se limitan a ningún estándar de codificación particular.

[0052] 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, encriptados, decodificados o utilizados de otro modo en el proceso de encriptado y/o decodificación). Por ejemplo, un bloque puede incluir una matriz bidimensional de muestras de datos de luminancia y/o crominancia. En general, el codificador de video

200 y el decodificador de video 300 pueden codificar datos de video representados en un formato YUV (por ejemplo, Y, Cb, Cr). Es decir, en lugar de codificar datos rojos, verdes y azules (RGB) para muestras de una imagen, el codificador de video 200 y el decodificador de video 300 pueden codificar componentes de luminancia y crominancia, donde los componentes de crominancia pueden incluir componentes de crominancia de tono rojo y azul. En algunos ejemplos, el codificador de video 200 convierte los datos con formato RGB recibidos en una representación YUV antes del encriptado, y el decodificador de video 300 convierte la representación YUV en el formato RGB. Alternativamente, las unidades de procesamiento previo y posterior (no mostradas) pueden realizar estas conversiones.

[0053] Esta divulgación puede referirse generalmente a la codificación (por ejemplo, encriptado y decodificación) de imágenes para incluir el proceso de encriptado 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 encriptado 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.

[0054] El HEVC define varios bloques, que incluyen unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). De acuerdo con HEVC, un codificador de video (como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU según una estructura quadtree. Es decir, el codificador de video divide las CTU y CU en cuatro cuadrados iguales y no superpuestos, y cada nodo del quadtree tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse “nodos hoja”, y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de video puede particionar más las PU y TU. Por ejemplo, en HEVC, un quadtree residual (RQT) representa la partición de las 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 de intrapredicción.

[0055] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para funcionar de acuerdo con JEM o VVC. Según JEM o VVC, un codificador de video (como el codificador de video 200) divide una imagen en una

pluralidad de CTU. El codificador de video 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol binario quadtree (QTBT) o una estructura de árbol de tipo múltiple (MTT). La estructura QTBT elimina los conceptos de múltiples tipos de partición, como la separación entre CU, PU y TU de HEVC. Una estructura QTBT incluye dos niveles: un primer nivel particionado de acuerdo con la partición quadtree y un segundo nivel particionado de acuerdo con la partición de árbol binario. Un nodo raíz de la estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a unidades de codificación (CU).

[0056] En una estructura de partición MTT, los bloques pueden dividirse con el uso de una partición quadtree (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT). Una partición de árbol triple es una partición en la que un bloque se divide en tres sub-bloques. En algunos ejemplos, una partición de árbol triple divide un bloque en tres sub-bloques sin dividir el bloque original a través del centro. Los tipos de partición en MTT (por ejemplo, QT, BT y TT) pueden ser simétricos o asimétricos.

[0057] 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).

[0058] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar el particionamiento quadtree, particionamiento QTBT, particionamiento MTT u otras estructuras de particionamiento. Para fines de explicación, la descripción de las técnicas de la presente divulgación se presenta con respecto a la partición QTBT. Sin embargo, debe entenderse que las técnicas de la presente divulgación también pueden aplicarse a los codificadores de video configurados para usar la partición quadtree u otros tipos de partición.

[0059] La presente 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 tiene 16 muestras en dirección vertical ($y = 16$) y 16 muestras en dirección horizontal ($x = 16$). Del mismo modo, una CU de NxN tiene N muestras en dirección vertical y N muestras en dirección

horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden organizarse en filas y columnas. Además, las CU no necesitan necesariamente tener el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden incluir muestras de $N \times M$, donde M no es necesariamente igual a N .

[0060] El codificador de video 200 codifica datos de video para CU que representan predicción y/o información residual, y otra información. La información de predicción indica cómo se debe predecir la CU para formar un bloque de predicción para la CU. La información residual generalmente representa diferencias muestra por muestra entre muestras de CU antes del encriptado y el bloque de predicción.

[0061] 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 o más imágenes previamente codificadas, mientras que la intrapredicción generalmente se refiere a la predicción de la CU a partir de datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de predicción con el uso de uno o más vectores de movimiento. El codificador de video 200 generalmente puede realizar una búsqueda de movimiento para identificar un bloque de referencia que coincida estrechamente con la CU, por ejemplo, en términos de diferencias entre la CU y el bloque de referencia. El codificador de video 200 puede calcular una métrica de diferencia con el uso de una suma de diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), diferencia absoluta media (MAD), diferencias cuadradas medias (MSD) u otros cálculos de diferencia similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual con el uso de predicción unidireccional o predicción bidireccional.

[0062] Algunos ejemplos de JEM y VVC también proporcionan un modo de compensación de movimiento afín, que puede considerarse un modo de interpredicción. En el modo de compensación de movimiento afín, el codificador de video 200 puede determinar dos o más vectores de movimiento que representan movimiento no traslacional, tal como acercar o alejar, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0063] Para realizar la intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. Algunos ejemplos de JEM y VVC proporcionan sesenta y siete modos de intrapredicción, incluidos varios modos

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

[0064] En general, un codificador de video, tal como un codificador de video 200 o un decodificador de video 300, puede generar un bloque de predicción con el uso del modo planar como un gradiente estimado a partir de muestras vecinas. Por ejemplo, en VVC WD4, un codificador de video puede generar un bloque de predicción con el uso del modo de intrapredicción planar de la siguiente manera.

8.3.4.2.5 Especificación del modo de intrapredicción INTRA_PLANAR

Las entradas en este proceso son:

- una variable nTbW que especifica el ancho del bloque de transformación,
 - una variable nTbH que especifica la altura del bloque de transformación,
 - las muestras vecinas $p[x][y]$, con $x = -1$, $y = -1..nTbH$ y $x = 0..nTbW$, $y = -1$.
- Los resultados de este proceso son las muestras predichas $predSamples[x][y]$, con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

Las variables nW y nH se derivan de la siguiente manera:

$$nW = \text{Max}(nTbW, 2) \quad (8-115)$$

$$nH = \text{Max}(nTbH, 2) \quad (8-116)$$

Los valores de las muestras de predicción $predSamples[x][y]$, con $x = 0..nTbW - 1$ e $y = 0..nTbH - 1$, se derivan de la siguiente manera:

$$predV[x][y] = ((nH - 1 - y) * p[x][-1] + (y + 1) * p[-1][nTbH]) \ll \text{Log2}(nW) \quad (8-117)$$

$$predH[x][y] = ((nW - 1 - x) * p[-1][y] + (x + 1) * p[nTbW][-1]) \ll \text{Log2}(nH) \quad (8-118)$$

$$predSamples[x][y] = (predV[x][y] + predH[x][y] + nW * nH) \gg (\text{Log2}(nW) + \text{Log2}(nH) + 1) \quad (8-119)$$

[0065] El codificador de video 200 codifica datos que representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de video 200 puede codificar datos que representan cuál de los diversos modos de interpredicción disponibles se usa, así como información de movimiento para el modo

correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador de video 200 puede codificar vectores de movimiento con el uso de la predicción avanzada de vectores de movimiento (AMVP) o el modo de fusión. El codificador de video 200 puede usar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afín.

[0066] Después de la predicción, tal como la intrapredicción o la interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para el bloque. Los datos residuales, como un bloque residual, representan diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado con el uso del modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformadas al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada discreta de coseno (DCT), una transformada entera, una transformada wavelet o una transformada conceptualmente similar a los datos de video residuales. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformación, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada Karhunen-Loeve (KLT) o similar. El codificador de video 200 produce coeficientes de transformación después de la aplicación de una o más transformadas.

[0067] Como se indicó con anterioridad, después de cualquier transformada para producir coeficientes de transformación, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en donde los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes de transformación, proporcionando así una compresión adicional. Al realizar el proceso de cuantificación, el codificador de video 200 puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación. Por ejemplo, el codificador de video 200 puede redondear un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación, el codificador de video 200 puede realizar un desplazamiento a la derecha bit a bit del valor a cuantificar.

[0068] Después de la cuantificación, el codificador de video 200 puede explorar los coeficientes de transformación, produciendo así un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La

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

[0069] Como se mencionó con anterioridad, el codificador de video 200 y el decodificador de video 300 pueden aplicar encriptado y decodificación CABAC a valores de elementos de sintaxis. Para aplicar el encriptado CABAC a un elemento de sintaxis, el codificador de video 200 puede binarizar el valor del elemento de sintaxis para formar una serie de uno o más bits, que se denominan “bins”. Además, el codificador de video 200 puede identificar un contexto de codificación, que puede denominarse simplemente “contexto”. El contexto de codificación puede identificar probabilidades de que los bins tengan valores particulares. Por ejemplo, un contexto de codificación puede indicar una probabilidad de 0,7 de codificar un bin de valor 0 y una probabilidad de 0,3 de codificar un bin de valor 1. Después de identificar el contexto de codificación, el codificador de video 200 puede dividir un intervalo en un subintervalo inferior y un subintervalo superior. Uno de los subintervalos puede estar asociado con el valor 0 y el otro subintervalo puede estar asociado con el valor 1. Los anchos de los subintervalos pueden ser proporcionales a las probabilidades indicadas para los valores asociados por el contexto de codificación identificado. Si un bin del elemento de sintaxis tiene el valor asociado con el subintervalo inferior, el valor codificado puede ser igual al límite inferior del subintervalo inferior. Si el mismo bin del elemento de sintaxis tiene el valor asociado con el subintervalo superior, el valor codificado puede ser igual al límite inferior del subintervalo superior. Para codificar el siguiente bin del elemento de sintaxis, el

codificador de video 200 puede repetir estos pasos siendo el intervalo el subintervalo asociado con el valor del bit codificado. Cuando el codificador de video 200 repite estos pasos para el siguiente bin, el codificador de video 200 puede usar probabilidades modificadas en función de las probabilidades indicadas por el contexto de codificación identificado y los valores reales de los bins codificados.

[0070] Cuando el decodificador de video 300 realiza la decodificación CABAC en un valor de un elemento de sintaxis, el decodificador de video 300 puede identificar un contexto de codificación. El decodificador de video 300 puede entonces dividir un intervalo en un subintervalo inferior y un subintervalo superior. Uno de los subintervalos puede estar asociado con el valor 0 y el otro subintervalo puede estar asociado con el valor 1. Los anchos de los subintervalos pueden ser proporcionales a las probabilidades indicadas para los valores asociados por el contexto de codificación identificado. Si el valor codificado está dentro del subintervalo inferior, el decodificador de video 300 puede decodificar un bin que tiene el valor asociado con el subintervalo inferior. Si el valor codificado está dentro del subintervalo superior, el decodificador de video 300 puede decodificar un bin que tiene el valor asociado con el subintervalo superior. Para decodificar un bin siguiente del elemento de sintaxis, el decodificador de video 300 puede repetir estos pasos siendo el intervalo el subintervalo que contiene el valor codificado. Cuando el decodificador de video 300 repite estos pasos para el siguiente bin, el decodificador de video 300 puede usar probabilidades modificadas en función de las probabilidades indicadas por el contexto de codificación identificado y los bins decodificados. El decodificador de video 300 puede entonces desbinarizar los bins para recuperar el valor del elemento de sintaxis.

[0071] En algunos casos, el codificador de video 200 puede codificar bins con el uso de la codificación CABAC de derivación. Puede resultar computacionalmente menos costoso realizar la codificación CABAC de derivación en un bin que realizar la codificación CABAC normal en el bin. Además, realizar la codificación CABAC de derivación puede permitir un mayor grado de paralelización y rendimiento. Los bins codificados mediante la codificación CABAC de derivación pueden denominarse “bins de derivación”. La agrupación de bins de derivación puede aumentar el rendimiento del codificador de video 200 y el decodificador de video 300. El motor de codificación CABAC de derivación puede codificar varios bins en un solo ciclo, mientras que el motor de codificación CABAC normal puede codificar solo un bin en un solo ciclo. El motor de codificación CABAC de derivación puede ser más simple porque el motor de codificación CABAC de derivación no selecciona contextos y puede asumir una probabilidad de $\frac{1}{2}$

para ambos símbolos (0 y 1). En consecuencia, en la codificación CABAC de derivación, los intervalos se dividen directamente por la mitad.

[0072] 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, 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.

[0073] De esta manera, el codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados, por ejemplo, elementos de sintaxis que describen la partición de una imagen en bloques (por ejemplo, CU) e información de predicción y/o residual para los bloques. Finalmente, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[0074] En general, el decodificador de video 300 realiza un proceso recíproco al realizado por el codificador de video 200 para decodificar los datos de video codificados del flujo de bits. Por ejemplo, el decodificador de video 300 puede decodificar valores para elementos de sintaxis del flujo de bits con el uso de CABAC de una manera sustancialmente similar, aunque recíproca, al proceso de encriptado de CABAC del codificador de video 200. Los elementos de sintaxis pueden definir información de partición de una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, como una estructura QTBT, para definir las CU de la CTU. Los elementos de sintaxis pueden definir aún más la información de predicción y la residual para bloques (por ejemplo, CU) de datos de video.

[0075] La información residual puede estar representada, por ejemplo, por coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantificar y transformar inversamente los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 usa un modo de predicción señalado (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.

[0076] La intrapredicción implica un modo de predicción DC, un modo de predicción planar y modos de predicción direccional (o angular). La predicción direccional para bloques cuadrados utiliza 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)*”, 11ª reunión de JVET, Liubliana, SI, julio de 2018, JVET-K1002), como se ilustra en la Figura 2. La Figura 2 es un diagrama de bloques que ilustra un ejemplo de intrapredicción, con flechas que apuntan hacia muestras de referencia.

[0077] En VTM2, la estructura de bloque usada 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 la codificación en función de las características del contenido.

[0078] En tales bloques rectangulares, restringir la dirección de la intrapredicción para que esté entre -135 grados y 45 grados puede dar como resultado situaciones en las que se utilizan 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; es más beneficioso relajar el intervalo de restricciones para que las muestras de referencia más cercanas (más allá del ángulo de -135 a 45 grados) puedan usarse para la predicción. Un ejemplo de tal caso se da en la Figura 3. La Figura 3 es un diagrama conceptual de un ejemplo de bloque rectangular de 8x4 en donde no se utilizan muestras de referencia “más cercanas” (círculo punteado superior derecho 502), pero se pueden utilizar muestras de referencia más lejanas (círculo punteado inferior izquierdo 504) debido a una restricción de una dirección de intrapredicción para que esté en el intervalo de -135 grados a 45 grados. La Figura 4 es una ilustración de ángulos amplios que se adoptan en VTM2.

[0079] Durante la 12ª reunión de JVET, se adoptó una modificación de la intrapredicción de ángulo amplio en VTM3 (ver L. Zhao, X. Zhao, S. Liu, X. Li, “*CE3-related: Unification of angular intra prediction for square and non-square blocks*”, 12ª reunión de JVET, RAE de Macao, CN, octubre de 2018, JVET-L0279; J. Chen, Y. Ye, S. Kim, “*Algorithm description for Versatile Video Coding and Test Model 3 (VTM3)*”, 12ª reunión de JVET, RAE de Macao, CN, octubre de 2018, JVET-L1002; B. Bross, J. Chen, S. Liu, “*Versatile Video Coding (Draft 3)*”, 12ª reunión de JVET, RAE de Macao, CN, octubre de 2018, JVET-L1001). Esta adopción incluye dos modificaciones para unificar la intrapredicción angular para bloques cuadrados y no cuadrados. En primer lugar, las direcciones de

predicción angular se modifican para cubrir las direcciones diagonales de todas las formas de bloque. En segundo lugar, todas las direcciones angulares se mantienen dentro del intervalo entre la dirección diagonal inferior izquierda y la dirección diagonal superior derecha para todas las relaciones de aspecto de bloque (cuadradas y no cuadradas) como se ilustra en las Figuras 5A-5C. Las Figuras 5A, 5B y 5C son ilustraciones de procesos de mapeo de modo para modos de intrapredicción fuera de un intervalo de dirección diagonal. La Figura 5A ilustra un bloque cuadrado que no requiere remapeo de modo angular. La Figura 5B ilustra el remapeo de modo angular para un bloque horizontal no cuadrado. La Figura 5C ilustra el remapeo de modo angular para un bloque vertical no cuadrado.

[0080] Además, el número de muestras de referencia en la fila de referencia superior y la columna de referencia izquierda está restringido a $2 * \text{ancho} + 1$ y $2 * \text{altura} + 1$ para todas las formas de bloque. Una ilustración de ángulos más amplios que se adoptan en VTM3 se proporciona en la Figura 6. La Figura 6 es un diagrama conceptual que ilustra ángulos amplios (-1 a -14 y 67 a 80) en el modelo de prueba 3 de VVC (VTM3) más allá de los modos 2 y 66 para un total de 93 modos angulares. Aunque VTM3 define 95 modos, para cualquier tamaño de bloque solo se permiten 67 modos. Los modos exactos que están permitidos dependen de la relación entre el ancho y la altura del bloque. Esto se hace al restringir el intervalo de modo para ciertos tamaños de bloques.

[0081] La Tabla 1, mostrada en la Figura 7, especifica la tabla de mapeo entre un modo de intrapredicción (predModelIntra) y el parámetro de ángulo intraPredAngle en VTM3 (JVET-L1001). Los modos angulares correspondientes a diagonales de bloques no cuadrados se indican con subrayado. Los modos vertical y horizontal se indican en cursiva. Los modos diagonales de bloques cuadrados se indican con subrayado y cursiva. Los modos diagonales de un bloque tienen cada uno un ángulo igual al ángulo de una línea entre las esquinas inferior izquierda y superior derecha del bloque o las esquinas superior izquierda e inferior derecha del bloque. En esta divulgación, 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).

[0082] El parámetro de ángulo inverso invAngle se deriva en función de intraPredAngle de la siguiente manera:

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

[0083] Obsérvese que los valores intraPredAngle que son múltiplos de 32 (0, 32, 64,

128, 256, 512) se corresponden con la predicción de muestras de matriz de referencia no fraccionarias, como es el caso en la especificación VTM3.

[0084] La Tabla 2, a continuación, muestra modos diagonales (es decir, modos de intrapredicción diagonales) para bloques que tienen varias relaciones de aspecto.

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

Relación de aspecto del 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

[0085] Las muestras en la vecindad de un bloque de codificación se utilizan para la intrapredicción del bloque. Normalmente, las líneas de muestra de referencia reconstruidas que están más cercanas a los límites izquierdo y superior del bloque de codificación se utilizan como muestras de referencia para la intrapredicción. Sin embargo, VVC WD4 también permite que otras muestras en la vecindad del bloque de codificación se utilicen como muestras de referencia. La Figura 8 ilustra las líneas de muestra de referencia que pueden usarse para la intrapredicción. En el ejemplo de la Figura 8, las líneas de muestra de referencia se indican mediante un índice de línea de referencia múltiple (MRLIdx). Para cada bloque de codificación se señala un índice que indica la línea de referencia que se utiliza.

[0086] El valor del índice señalado puede ser diferente del MRLIdx indicado por el índice señalado. Por ejemplo, en VVC WD4, solo se pueden usar líneas de referencia con MRLIdx igual a 0, 1 y 3. En VVC WD4, el índice señalado a la línea de referencia utilizada 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 planar y DC no se utilizan cuando MRLIdx > 0. En una versión más reciente de VVC, se puede utilizar MRLIdx igual a 0, 1 y 2, y se pueden utilizar modos DC cuando MRLIdx > 0.

[0087] Para hacer más eficiente el proceso de señalización de modos de intrapredicción, el codificador 200 de video y el decodificador de video 300 pueden identificar uno o más modos de intrapredicción como los “modos más probables” (MPM)

para un bloque. Si el modo de intrapredicción del bloque es uno de los MPM del bloque, se señala un índice al MPM. Si el modo de intrapredicción del bloque no es uno de los MPM para el bloque, se puede señalar un índice para uno de los modos de intrapredicción no MPM para el bloque. Debido a que normalmente hay muchos menos MPM para un bloque que los modos de intrapredicción que no son MPM, el índice del MPM se puede señalar con el uso de menos bits que el índice de un modo de intrapredicción que no es MPM. El siguiente texto de VVC WD4 en un ejemplo de un proceso para determinar un modo de intrapredicción que determina MPM para un bloque.

Las entradas en este proceso son:

una ubicación de luma (x_{Cb} , y_{Cb}) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior izquierda de la imagen actual,

una variable $cbWidth$ que especifica el ancho del bloque de codificación actual en muestras de luma,

una variable $cbHeight$ que especifica la altura del bloque de codificación actual en muestras de luma.

En este proceso, se deriva el modo de intrapredicción de luma $IntraPredModeY[x_{Cb}][y_{Cb}]$.

La Tabla 8-1 especifica el valor para el modo de intrapredicción $IntraPredModeY[x_{Cb}][y_{Cb}]$ y los nombres asociados.

Tabla 8-1 - Especificación del modo de intrapredicción y nombres asociados

Modo de intrapredicción	Nombre asociado
0	INTRA_PLANAR
1	INTRA_DC
2..66	INTRA_ANGULAR2..INTRA_ANGULAR66
81..83	INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM

NOTA - : Los modos de intrapredicción INTRA_LT_CCLM, INTRA_L_CCLM e INTRA_T_CCLM solo son aplicables a los componentes cromáticos.

$IntraPredModeY[x_{Cb}][y_{Cb}]$ se deriva mediante los siguientes pasos ordenados:

Las ubicaciones vecinas (x_{NbA} , y_{NbA}) y (x_{NbB} , y_{NbB}) se establecen igual a ($x_{Cb} - 1$, $y_{Cb} + cbHeight - 1$) y ($x_{Cb} + cbWidth - 1$, $y_{Cb} - 1$), respectivamente.

Para que X sea reemplazado por A o B, las variables $candIntraPredModeX$ se derivan de la siguiente manera:

El proceso de derivación de disponibilidad para un bloque como se especifica en la cláusula 6.4.X [Ed. (BB): proceso de verificación de disponibilidad de bloques vecinos tbd] se invoca con la ubicación (xCurr, yCurr) igual a (xCb, yCb) y la ubicación vecina (xNbY, yNbY) igual a (xNbX, yNbX) como entradas, y la salida se asigna a availableX.

El modo candidato de intrapredicción candIntraPredModeX se deriva de la siguiente manera:

Si una o más de las siguientes condiciones son verdaderas, candIntraPredModeX se establece igual a INTRA_PLANAR.

La variable availableX es igual a FALSE.

CuPredMode[xNbX][yNbX] no es igual a MODE_INTRA y ciip_flag[xNbX][yNbX] no es igual a 1.

pcm_flag[xNbX][yNbX] es igual a 1.

X es igual a B e yCb - 1 es menor que ((yCb >> CtbLog2SizeY) << CtbLog2SizeY).

De lo contrario, candIntraPredModeX se establece igual a IntraPredModeY[xNbX][yNbX].

Las variables ispDefaultMode1 e ispDefaultMode2 se definen de la siguiente manera:

Si IntraSubPartitionsSplitType es igual a ISP_HOR_SPLIT, ispDefaultMode1 se establece igual a INTRA_ANGULAR18 e ispDefaultMode2 se establece igual a INTRA_ANGULAR5.

De lo contrario, ispDefaultMode1 se establece igual a INTRA_ANGULAR50 e ispDefaultMode2 se establece igual a INTRA_ANGULAR63.

CandModeList [x] con x = 0..5 se deriva de la siguiente manera:

Si candIntraPredModeB es igual a candIntraPredModeA y candIntraPredModeA es mayor que INTRA_DC, candModeList [x] con x = 0..5 se deriva de la siguiente manera:

Si IntraLumaRefLineIdx [xCb][yCb] es igual a 0 e IntraSubPartitionsSplitType es igual a ISP_NO_SPLIT, se aplica lo siguiente:

candModeList[0] = candIntraPredModeA (8-8)

candModeList[1] = INTRA_PLANAR (8-9)

candModeList[2] = INTRA_DC (8-10)

candModeList[3] = 2 + ((candIntraPredModeA + 61) % 64) (8-11)

candModeList[4] = 2 + ((candIntraPredModeA - 1) % 64) (8-12)

candModeList[5] = 2 + ((candIntraPredModeA + 60) % 64) (8-13)

De lo contrario ($\text{IntraLumaRefLineIdx}[x_{Cb}][y_{Cb}]$ no es igual a 0 o $\text{IntraSubPartitionsSplitType}$ no es igual a ISP_NO_SPLIT), se aplica lo siguiente:

$$\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-14)$$

$$\text{candModeList}[1] = 2 + ((\text{candIntraPredModeA} + 61) \% 64) \quad (8-15)$$

$$\text{candModeList}[2] = 2 + ((\text{candIntraPredModeA} - 1) \% 64) \quad (8-16)$$

Si se cumple una de las siguientes condiciones,

$\text{IntraSubPartitionsSplitType}$ es igual a ISP_HOR_SPLIT y $\text{candIntraPredModeA}$ es menor que INTRA_ANGULAR34 ,

$\text{IntraSubPartitionsSplitType}$ es igual a ISP_VER_SPLIT y $\text{candIntraPredModeA}$ es mayor o igual que INTRA_ANGULAR34 ,

$\text{IntraLumaRefLineIdx}[x_{Cb}][y_{Cb}]$ no es igual a 0,

se aplica lo siguiente:

$$\text{candModeList}[3] = 2 + ((\text{candIntraPredModeA} + 60) \% 64) \quad (8-17)$$

$$\text{candModeList}[4] = 2 + (\text{candIntraPredModeA} \% 64) \quad (8-18)$$

$$\text{candModeList}[5] = 2 + ((\text{candIntraPredModeA} + 59) \% 64) \quad (8-19)$$

De lo contrario, se aplica lo siguiente:

$$\text{candModeList}[3] = \text{ispDefaultMode1} \quad (8-20)$$

$$\text{candModeList}[4] = \text{ispDefaultMode2} \quad (8-21)$$

$$\text{candModeList}[5] = \text{INTRA\_PLANAR} \quad (8-22)$$

De lo contrario, si $\text{candIntraPredModeB}$ no es igual a $\text{candIntraPredModeA}$ y $\text{candIntraPredModeA}$ o $\text{candIntraPredModeB}$ es mayor que INTRA_DC , se aplica lo siguiente:

Las variables minAB y maxAB se derivan de la siguiente manera:

$$\text{minAB} = \text{Min}(\text{candIntraPredModeA}, \text{candIntraPredModeB}) \quad (8-23)$$

$$\text{maxAB} = \text{Max}(\text{candIntraPredModeA}, \text{candIntraPredModeB}) \quad (8-24)$$

Si $\text{candIntraPredModeA}$ y $\text{candIntraPredModeB}$ son mayores que INTRA_DC , $\text{candModeList}[x]$ con $x = 0..5$ se deriva de la siguiente manera:

$$\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-25)$$

$$\text{candModeList}[1] = \text{candIntraPredModeB} \quad (8-26)$$

Si $\text{IntraLumaRefLineIdx}[x_{Cb}][y_{Cb}]$ es igual a 0 e $\text{IntraSubPartitionsSplitType}$ es igual a ISP_NO_SPLIT , se aplica lo siguiente:

$$\text{candModeList}[2] = \text{INTRA\_PLANAR} \quad (8-27)$$

$$\text{candModeList}[3] = \text{INTRA\_DC} \quad (8-28)$$

Si $\text{maxAB} - \text{minAB}$ está en el intervalo de 2 a 62, inclusive, se aplica lo siguiente:

$$\text{candModeList}[4] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-29)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-30)$$

De lo contrario, se aplica lo siguiente:

$$\text{candModeList}[4] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-31)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB}) \% 64) \quad (8-32)$$

De lo contrario ($\text{IntraLumaRefLineIdx}[\text{xCb}][\text{yCb}]$ no es igual a 0 o $\text{IntraSubPartitionsSplitType}$ no es igual a ISP_NO_SPLIT), se aplica lo siguiente:

Cuando $\text{IntraSubPartitionsSplitType}$ no es igual a ISP_NO_SPLIT y $\text{abs}(\text{candIntraPredModeB} - \text{ispDefaultMode1})$ es menor que $\text{abs}(\text{candIntraPredModeA} - \text{ispDefaultMode1})$, se aplica lo siguiente:

$$\text{candModeList}[0] = \text{candIntraPredModeB} \quad (8-33)$$

$$\text{candModeList}[1] = \text{candIntraPredModeA} \quad (8-34)$$

Si $\text{maxAB} - \text{minAB}$ es igual a 1, se aplica lo siguiente:

$$\text{candModeList}[2] = 2 + ((\text{minAB} + 61) \% 64) \quad (8-35)$$

$$\text{candModeList}[3] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-36)$$

$$\text{candModeList}[4] = 2 + ((\text{minAB} + 60) \% 64) \quad (8-37)$$

$$\text{candModeList}[5] = 2 + (\text{maxAB} \% 64) \quad (8-38)$$

De lo contrario, si $\text{maxAB} - \text{minAB}$ es igual a 2, se aplica lo siguiente:

$$\text{candModeList}[2] = 2 + ((\text{minAB} - 1) \% 64) \quad (8-39)$$

$$\text{candModeList}[3] = 2 + ((\text{minAB} + 61) \% 64) \quad (8-40)$$

$$\text{candModeList}[4] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-41)$$

$$\text{candModeList}[5] = 2 + ((\text{minAB} + 60) \% 64) \quad (8-42)$$

De lo contrario, si $\text{maxAB} - \text{minAB}$ es mayor que 61, se aplica lo siguiente:

$$\text{candModeList}[2] = 2 + ((\text{minAB} - 1) \% 64) \quad (8-43)$$

$$\text{candModeList}[3] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-44)$$

$$\text{candModeList}[4] = 2 + (\text{minAB} \% 64) \quad (8-45)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-46)$$

De lo contrario, se aplica lo siguiente:

$$\text{candModeList}[2] = 2 + ((\text{minAB} + 61) \% 64) \quad (8-47)$$

$$\text{candModeList}[3] = 2 + ((\text{minAB} - 1) \% 64) \quad (8-48)$$

$$\text{candModeList}[4] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-49)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-50)$$

De lo contrario ($\text{candIntraPredModeA}$ o $\text{candIntraPredModeB}$ es mayor que INTRA_DC), $\text{candModeList}[x]$ con $x = 0..5$ se deriva de la siguiente manera:

Si $\text{IntraLumaRefLineIdx}[\text{xCb}][\text{yCb}]$ es igual a 0 e $\text{IntraSubPartitionsSplitType}$ es igual a ISP_NO_SPLIT , se aplica lo siguiente:

$$\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-51)$$

$$\text{candModeList}[1] = \text{candIntraPredModeB} \quad (8-52)$$

$$\text{candModeList}[2] = 1 - \text{minAB} \quad (8-53)$$

$$\text{candModeList}[3] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-54)$$

$$\text{candModeList}[4] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-55)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-56)$$

De lo contrario, si $\text{IntraLumaRefLineIdx}[\text{xCb}][\text{yCb}]$ no es igual a 0, se aplica lo siguiente:

$$\text{candModeList}[0] = \text{maxAB} \quad (8-57)$$

$$\text{candModeList}[1] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-58)$$

$$\text{candModeList}[2] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-59)$$

$$\text{candModeList}[3] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-60)$$

$$\text{candModeList}[4] = 2 + (\text{maxAB} \% 64) \quad (8-61)$$

$$\text{candModeList}[5] = 2 + ((\text{maxAB} + 59) \% 64) \quad (8-62)$$

De lo contrario ($\text{IntraSubPartitionsSplitType}$ no es igual a ISP_NO_SPLIT), se aplica lo siguiente:

$$\text{candModeList}[0] = \text{INTRA\_PLANAR} \quad (8-63)$$

$$\text{candModeList}[1] = \text{maxAB} \quad (8-64)$$

$$\text{candModeList}[2] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-65)$$

$$\text{candModeList}[3] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-66)$$

$$\text{candModeList}[4] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-67)$$

$$\text{candModeList}[5] = 2 + (\text{maxAB} \% 64) \quad (8-68)$$

De lo contrario, se aplica lo siguiente:

Si $\text{IntraLumaRefLineIdx}[\text{xCb}][\text{yCb}]$ es igual a 0 e $\text{IntraSubPartitionsSplitType}$ es igual a ISP_NO_SPLIT , se aplica lo siguiente:

$$\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-69)$$

$$\text{candModeList}[1] = (\text{candModeList}[0] == \text{INTRA\_PLANAR}) ? \text{INTRA\_DC} : \text{INTRA\_PLANAR} \quad (8-70)$$

$$\text{candModeList}[2] = \text{INTRA\_ANGULAR50} \quad (8-71)$$

$$\text{candModeList}[3] = \text{INTRA\_ANGULAR18} \quad (8-72)$$

$$\text{candModeList}[4] = \text{INTRA\_ANGULAR46} \quad (8-73)$$

$$\text{candModeList}[5] = \text{INTRA\_ANGULAR54} \quad (8-74)$$

De lo contrario, si $\text{IntraLumaRefLineIdx}[\text{xCb}][\text{yCb}]$ no es igual a 0, se aplica lo siguiente:

$$\text{candModeList}[0] = \text{INTRA\_ANGULAR50} \quad (8-75)$$

candModeList[1] = INTRA_ANGULAR18 (8-76)

candModeList[2] = INTRA_ANGULAR2 (8-77)

candModeList[3] = INTRA_ANGULAR34 (8-78)

candModeList[4] = INTRA_ANGULAR66 (8-79)

candModeList[5] = INTRA_ANGULAR26 (8-80)

De lo contrario, si IntraSubPartitionsSplitType es igual a ISP_HOR_SPLIT, se aplica lo siguiente:

candModeList[0] = INTRA_PLANAR (8-81)

candModeList[1] = INTRA_ANGULAR18 (8-82)

candModeList[2] = INTRA_ANGULAR25 (8-83)

candModeList[3] = INTRA_ANGULAR10 (8-84)

candModeList[4] = INTRA_ANGULAR65 (8-85)

candModeList[5] = INTRA_ANGULAR50 (8-86)

De lo contrario, si IntraSubPartitionsSplitType es igual a ISP_VER_SPLIT, se aplica lo siguiente:

candModeList[0] = INTRA_PLANAR (8-87)

candModeList[1] = INTRA_ANGULAR50 (8-88)

candModeList[2] = INTRA_ANGULAR43 (8-89)

candModeList[3] = INTRA_ANGULAR60 (8-90)

candModeList[4] = INTRA_ANGULAR3 (8-91)

candModeList[5] = INTRA_ANGULAR18 (8-92)

IntraPredModeY[xCb][yCb] se deriva al aplicar el siguiente procedimiento:

Si intra_luma_mpm_flag[xCb][yCb] es igual a 1, IntraPredModeY[xCb][yCb] se establece igual a candModeList[intra_luma_mpm_idx[xCb][yCb]].

De lo contrario, IntraPredModeY[xCb][yCb] se deriva al aplicar los siguientes pasos ordenados:

Cuando candModeList[i] es mayor que candModeList[j] para i = 0..4 y para cada i, j = (i + 1)..5, ambos valores se intercambian de la siguiente manera:

(candModeList[i], candModeList[j]) = Swap(candModeList[i], candModeList[j]) (8-93)

IntraPredModeY[xCb][yCb] se deriva mediante los siguientes pasos ordenados:

IntraPredModeY[xCb][yCb] es igual a intra_luma_mpm_remainder[xCb][yCb].

Para i igual a 0 a 5, inclusive, cuando IntraPredModeY[xCb][yCb] es mayor o

igual que `candModeList [i]`, el valor de `IntraPredModeY[ xCb ][ yCb ]` se incrementa en uno.

La variable `IntraPredModeY [x] [y]` con $x = xCb..xCb + cbWidth - 1$ e $y = yCb.yCb + cbHeight - 1$ se establece para que sea igual a `IntraPredModeY[ xCb ][ yCb ]`.

[0088] La codificación intrasubpartición (ISP) (ver S. De Luxán Hernández, V. George, J. Ma, T. Nguyen, H. Schwarz, D. Marpe, T. Wiegand (HHI), “*CE3: Intra Sub-Partitions Coding Mode*”, JVET-M0102) es un método mediante el cual un bloque de codificación se divide en dos o cuatro sub-bloques. Cada sub-bloque dentro de un bloque se reconstruye en el orden de decodificación antes de la reconstrucción del sub-bloque subsiguiente en el orden de decodificación. En VVC WD4, ISP solo se aplica a bloques de codificación de luma. Las muestras de referencia para bloques codificados por ISP están restringidas a ser de la línea de referencia más cercana al bloque de codificación (por ejemplo, `MRLIdx = 0` como se muestra en la Figura 8). Se muestran ejemplos de división en la Figura 9 y Figura 10.

[0089] Se usa un bit para señalar si un bloque de codificación se divide en ISP y se usa un segundo bit para indicar el tipo de división: horizontal o vertical. Según el modo intra y el tipo de división utilizados, se pueden utilizar dos clases diferentes de órdenes de procesamiento, que se denominan orden normal e inverso. En el orden normal, la primera subpartición que se procesará es la que contiene la muestra superior izquierda de la CU y luego continúa hacia abajo (división horizontal) o hacia la derecha (división vertical). Por otro lado, el orden de procesamiento inverso comienza con la subpartición que contiene la muestra inferior izquierda de la CU (división horizontal) y continúa hacia arriba o comienza con la subpartición que contiene la muestra superior derecha de la CU y continúa hacia la izquierda (división vertical).

[0090] En JVET WD4 se usa una variación de ISP que usa solo el orden de procesamiento normal. Cabe señalar que los términos sub-bloque y subparticiones se utilizan indistintamente en este documento, y ambos se refieren a los bloques obtenidos mediante la partición de un bloque de codificación con el uso de ISP.

[0091] Algunas sintaxis y semánticas asociadas con ISP en JVET WD4 se muestran a continuación, con la sintaxis relevante encerrada en etiquetas `<!--... -->`

Tabla de sintaxis de la unidad de codificación

<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	Descriptor
<code>...</code>	
<code>} else {</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>if((y0 % CtbSizeY) > 0)</code>	

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 transformación

transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	Descriptor
<!-- 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 la unidad de codificación

<!=>intra_subpartitions_mode_flag[x0][y0] igual a 1 especifica que la unidad de intracodificación actual está dividida en subparticiones de bloque de transformación rectangular NumIntraSubPartitions[x0][y0]. **intra_subpartitions_mode_flag[x0][y0]** igual a 0 especifica que la unidad de intracodificación actual no está dividida en subparticiones de bloque de transformación 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 intra subparticiones 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 nombre 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 bloque de transformación 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 se cumple una de las siguientes condiciones, 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.<!/>

[0092] La intrapredicción lineal ponderada afín (ALWIP) (por ejemplo, como se describe en J. Pfaff *et al.*, “Affine linear weighted intra prediction (CE3-4.1, CE3-4.2)”, Equipo Conjunto de Expertos en Video (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC

29/WG 11, 14^a reunión: Ginebra, CH, 19-27 de marzo de 2019, documento JVET-N0217 (en adelante “JVET-N0217”)) genera una predicción de un bloque (es decir, un bloque de predicción) de las muestras de referencia vecinas con el uso de un modelo de predicción ponderada lineal afín. Las muestras vecinas se procesan primero (por ejemplo, en algunos casos, las muestras vecinas se muestrean de manera descendente) y las muestras procesadas (por ejemplo, muestreadas de manera descendente) se utilizan luego para derivar (con el uso de un modelo afín) un conjunto de muestras reducidas que se asemeja a una versión intermedia muestreada de manera descendente de las muestras predichas. Se obtiene una predicción final mediante el muestreo ascendente (según sea necesario) de los valores intermedios. Cabe destacar que ALWIP también puede denominarse intrapredicción de matriz (MIP).

[0093] En la Figura 11 se ofrece una ilustración del proceso ALWIP. El proceso ALWIP de la Figura 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 muestrean de manera descendente para obtener muestras de límite reducidas. La representación vectorial de las muestras de límite, $bdry_{red}$, se multiplica por una matriz A_k y se agrega un término de desviación/sesgo b_k para obtener una versión muestreada de manera descendente del bloque predicho, $pred_{red}$. La predicción final se obtiene mediante el muestreo ascendente de estas muestras predichas $pred_{red}$ junto con las muestras de los límites. La matriz A_k y un vector de desviación/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 desviación/sesgo b_k puede denominarse en el presente documento “modo ALWIP”.

[0094] La derivación de muestras predichas intermedias usa un modelo de predicción ponderada lineal afín. Se definen tres tipos (es decir, tipos ALWIP) y el número de muestras intermedias derivadas difiere para cada tipo de la siguiente manera:

- 1) 4x4 para tamaños de bloque de ancho y altura ambos iguales a 4. En otras palabras, hay 16 muestras intermedias para bloques que tienen tamaños de 4x4.
- 2) 8x8 para tamaños de bloque de ancho y altura, ambos menores que iguales a 8, excepto cuando tanto el ancho como la altura son iguales a 4 (es decir, bloques de 4x8, 8x4 y 8x8). En otras palabras, hay 64 muestras intermedias para bloques que tienen tamaños de $H \times W$, donde $4 < H \leq 8$ y $4 < W \leq 8$, o $H = 4$ y $4 < W \leq 8$, o $W = 4$ y $4 < H \leq 8$.
- 3) 16x16 para bloques donde al menos uno de ancho y altura es mayor que 8. En otras palabras, hay 256 muestras intermedias para bloques que tienen tamaños

de $H \times W$, ya sea $H > 8$ o $W > 8$.

En cada uno de estos tres casos, se utilizan diferentes números de modos ALWIP: 35, 19 y 11, respectivamente. En otras palabras, para ALWIP tipo 1, hay 35 modos ALWIP disponibles diferentes; para ALWIP tipo 2, hay 19 modos ALWIP disponibles diferentes; y para ALWIP tipo 3, hay 11 modos ALWIP diferentes disponibles. Cada uno de los diferentes modos ALWIP puede corresponder a una combinación diferente de una matriz A_k y un valor de desviación/sesgo b_k .

[0095] La señalización de la ALWIP incluye:

- a) Un indicador (`alwip_flag`) para indicar que el bloque actual está codificado con ALWIP.
- b) Cuando el bloque actual está codificado con ALWIP, se señala otro indicador para indicar si el bloque actual está codificado con un modo más probable (MPM) de ALWIP o no.
 - a. Si el bloque está codificado con un modo ALWIP-MPM, se señala un índice MPM. La derivación del modo ALWIP-MPM se describe en la presente divulgación a continuación.
 - b. De lo contrario, se señala un índice a un valor de modo restante. El valor de modo restante indica un modo ALWIP distinto del modo ALWIP-MPM.

[0096] El `alwip_flag` está codificado por contexto con cuatro contextos permitidos:

- Si el ancho del bloque $> 2 * \text{altura}$ o $\text{altura} > 2 * \text{ancho}$, se usa el contexto 3.
- De lo contrario, se utiliza el contexto `ctxId`, donde `ctxId` se deriva de la siguiente manera:
 - o Inicializar `ctxId` a 0.
 - o Si el bloque vecino izquierdo está codificado con ALWIP, `ctxId++` (es decir, incrementar `ctxId`).
 - o Si el bloque vecino de arriba está codificado con ALWIP, `ctxId++`.

[0097] La derivación de los modos ALWIP MPM implica los siguientes pasos:

- 1) `LeftIntraMode` y `AboveIntraMode` se inicializan en -1.
- 2) Si el bloque vecino izquierdo está intracodificado:
 - a. Si el bloque vecino izquierdo está codificado con el modo ALWIP L (es decir, si el bloque vecino izquierdo está codificado con ALWIP, dejar que el modo ALWIP del bloque vecino izquierdo se denote como L):
 - i. Si L es del mismo tipo ALWIP que el bloque actual, `LeftIntraMode` se establece igual a L .
 - b. De lo contrario, si el bloque vecino izquierdo no está codificado con un

modo ALWIP, el modo intra 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 de arriba está intracodificado:
 - a. Si el bloque vecino de arriba está codificado con el modo ALWIP A (es decir, si el bloque vecino de arriba está codificado con ALWIP, dejar que el modo ALWIP del bloque vecino de arriba se denote como A):
 - i. Si A es del mismo tipo ALWIP que el bloque actual, AboveIntraMode se establece igual a A .
 - b. De lo contrario, si el bloque vecino de arriba no está codificado con un modo ALWIP, el modo intra del bloque vecino de arriba se mapea a un modo ALWIP del mismo tipo que el bloque actual y se asigna a AboveIntraMode.
- 4) Los MPM se derivan luego en función de LeftIntraMode y AboveIntraMode. En otras palabras, un primer ALWIP-MPM se establece igual a LeftIntraMode y un segundo ALWIP-MPM se establece igual a AboveIntraMode. Si LeftIntraMode permanece igual a -1 o AboveIntraMode permanece igual a -1, los MPM correspondientes no están disponibles.

En el resto del documento, los bloques codificados con ALWIP pueden denominarse bloques codificados ALWIP o bloques ALWIP; otros bloques (por ejemplo, bloques codificados con intrapredicción regular, subparticiones internas o múltiples líneas de referencia) pueden denominarse bloques no ALWIP.

[0098] Existen varios problemas en el diseño de ALWIP (es decir, MIP). Por ejemplo, la derivación de contexto del indicador que indica si un bloque actual está codificado con el uso del modo ALWIP (por ejemplo, `alwip_flag`) usa los valores `alwip_flag` de los bloques vecinos izquierdo y de arriba. El valor de un `alwip_flag` del bloque vecino de arriba se utiliza incluso cuando el bloque vecino de arriba pertenece a una fila de CTU diferente. Esto crea requisitos de almacenamiento adicionales. Por ejemplo, si r se refiere al límite de CTU entre la fila de CTU actual y la fila de CTU anterior, es necesario almacenar el `alwip_flag` de todos los bloques de la fila de CTU anterior que comparte un límite r . En el peor de los casos de bloques 4x4 en una imagen de 4K, el número de bytes necesarios sería $3840/4$ (número de bloques) $\times$ 1 (bits por valor de indicador) = 120 bytes. Proporcionar almacenamiento para los valores `alwip_flag` de todos los bloques de la fila de CTU anterior aumenta los requisitos de almacenamiento de memoria del codificador de video 200 y el decodificador de video 300, lo que aumenta

potencialmente el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0099] En otro ejemplo de un problema con el diseño de ALWIP, hay 35 matrices/modos especificados para bloques 4x4. Estas 35 matrices/modos representan casi el 25 % de los parámetros necesarios para ALWIP. La eficiencia de compresión al habilitar bloques ALWIP de 4x4 es de casi el 0,05 % y el almacenamiento requerido alrededor de 2 KB (para almacenar las 35 matrices y los vectores de desviación/sesgo). Por lo tanto, la carga de almacenamiento en las implementaciones de codificador/decodificador es indebida para la eficiencia obtenida. La Figura 12 es una tabla que ilustra el número de parámetros de matriz y parámetros de desviación para cada tipo de ALWIP en VVC WD4.

[0100] Durante la derivación del modo más probable (MPM) de bloques no ALWIP, el modo intra utilizado para los bloques vecinos izquierdo y superior se comprueba y se utiliza cuando está disponible. Por ejemplo, al derivar los modos ALWIP MPM, un codificador de video (por ejemplo, codificador de video 200 y decodificador de video 300) determina si el bloque vecino izquierdo está disponible y, si es así, el codificador de video determina si el bloque vecino izquierdo está intracodificado y, si es así, el codificador de video determina si el bloque vecino izquierdo está codificado o no con el uso de un modo ALWIP. De manera similar, el codificador de video también determina si el bloque vecino de arriba está disponible y, si es así, el codificador de video determina si el bloque vecino de arriba está intracodificado y, si es así, el codificador de video determina si el bloque vecino de arriba está codificado o no con el uso de un modo ALWIP. Para bloques que no son ALWIP (es decir, bloques no codificados con el uso de un modo ALWIP), los valores de modo corresponden a los modos de intrapredicción angular o modos de intrapredicción Planar/DC. Excepto en casos triviales, un conjunto de coeficientes de matriz (por ejemplo, un conjunto de coeficientes de matriz para un modo ALWIP) utilizado en ALWIP puede no corresponder a ninguno de los modos intra utilizados en bloques que no son ALWIP. ALWIP utiliza actualmente una primera tabla de mapeo para aproximar el mapeo de la matriz ALWIP a uno de los modos intra. Este valor aproximado se considera entonces como el modo intra del bloque vecino que se utilizará en la derivación MPM del bloque no ALWIP. En otras palabras, cuando un bloque vecino izquierdo se codifica con el uso de un modo de intrapredicción no ALWIP, un codificador de video (por ejemplo, codificador de video 200 o decodificador de video 300) establece un primer ALWIP-MPM al valor aproximado para el bloque vecino izquierdo. De manera similar, cuando un bloque vecino de arriba se codifica con el uso

de un modo de intrapredicción no ALWIP, el codificador de video establece un segundo ALWIP-MPM en el valor aproximado para el bloque vecino de arriba. Esto puede resultar en una compresión ineficaz, por ejemplo, debido a un error en la aproximación. Además, la tabla de mapeo requiere almacenamiento adicional y agrega operaciones de búsqueda en el proceso de derivación de MPM. El almacenamiento adicional aumenta el costo y la complejidad del codificador de video 200 y el decodificador de video 300. Las operaciones de búsqueda adicionales pueden ralentizar los procesos de encriptado y decodificación.

[0101] Este problema también puede aplicarse al modo de modo directo (DM) de croma, en donde el modo intra utilizado para el bloque de croma se copia de un bloque intracontiguo. Cuando el intrabloque contiguo está codificado con ALWIP, la primera tabla de mapeo se usa para derivar el modo DM de croma a partir del modo/matriz de bloques ALWIP. En otras palabras, un bloque de luma contiguo con un bloque de croma se puede codificar con el uso de un modo ALWIP, pero el bloque de croma no se puede codificar con el uso de ningún modo de ALWIP. Por lo tanto, si el bloque de croma se va a codificar con el uso del modo DM de croma en donde el bloque de croma hereda el modo de intrapredicción del bloque de luma, puede ser necesario que un codificador de video (por ejemplo, encriptador de video 200 o decodificador de video 300) convierta el modo ALWIP del bloque de luma a un modo de intrapredicción no ALWIP para su uso en el bloque de croma. Esta conversión es una aproximación que puede reducir la eficiencia de la codificación. Además, esta conversión puede requerir almacenamiento adicional para una tabla de mapeo y puede agregar operaciones de búsqueda, lo que puede aumentar el costo y la complejidad del codificador de video y puede ralentizar los procesos de encriptado y decodificación.

[0102] De manera similar a los ejemplos de los párrafos anteriores, durante la derivación MPM de un bloque ALWIP, los modos intra de los bloques vecinos a la izquierda y arriba se comprueban y se usan cuando están disponibles. Cuando el bloque vecino es un bloque no ALWIP, el modo intra del bloque no ALWIP puede no corresponder con las matrices/modos usados por el bloque ALWIP. En otras palabras, los diferentes modos ALWIP disponibles para usar con un bloque actual dependiendo del tamaño del bloque actual y el modo intra de un bloque vecino superior o izquierdo del bloque actual pueden no corresponder a ninguno de los modos ALWIP disponibles para usar con el bloque actual. Por lo tanto, se usa una segunda tabla de mapeo para mapear el modo intra del bloque no ALWIP a la matriz/modo del bloque ALWIP. Esto puede resultar en una ineficacia en la compresión. Además, la segunda tabla de mapeo requiere

almacenamiento adicional y agrega operaciones de búsqueda en el proceso de derivación de MPM. Si un bloque vecino está codificado con modo intra/inter combinado (CIIP), el bloque vecino se considera (por ejemplo, por el codificador de video 200 y el decodificador de video 300) como un modo no intracodificado y se marca como no disponible. Si un bloque vecino está codificado con el modo de copia intrabloque (IBC), el bloque se considera un modo no intracodificado y se marca como no disponible.

[0103] La presente divulgación describe técnicas que pueden mejorar el diseño del modo ALWIP. Las técnicas de la presente divulgación pueden reducir el costo y la complejidad de los codificadores de video, como el codificador de video 200 y el decodificador de video 300, y pueden aumentar la velocidad de los procesos de encriptado y decodificación. Las técnicas y ejemplos de la presente divulgación pueden ser utilizados por el codificador de video 200 y el decodificador de video 300 individualmente o en cualquier combinación.

[0104] De acuerdo con una primera técnica de esta divulgación, cuando un bloque vecino de arriba pertenece a una CTU diferente a un bloque actual (es decir, un bloque que un codificador de video (por ejemplo, codificador de video 200 o decodificador de video 300) está codificando actualmente), se considera que el bloque vecino de arriba no está disponible a los efectos de derivar el contexto de un `alwip_flag`. Por ejemplo, un proceso de derivación del contexto para `alwip_flag` puede modificarse de la siguiente manera, con etiquetas `<!-->` que indican adiciones:

El `alwip_flag` está codificado por contexto con cuatro contextos permitidos:

- Si el ancho del bloque $> 2 * \text{altura}$ o $\text{altura} > 2 * \text{ancho}$, se usa el contexto 3.
- De lo contrario, se utiliza el contexto `ctxId`, donde `ctxId` se deriva de la siguiente manera:
 - o Inicializar `ctxId` a 0
 - o Si el bloque vecino izquierdo está codificado con ALWIP, `ctxId++`.
 - o Si el bloque vecino de arriba está codificado con ALWIP `<!-->` y el bloque vecino de arriba está en la misma CTU que el bloque actual `<!-->`, `ctxId++`

Por lo tanto, en este ejemplo, el codificador de video 200 y el decodificador de video 300 pueden evitar la necesidad de almacenar datos que indiquen si los bloques en una fila de CTU por encima de una fila de CTU que contiene el bloque actual están codificados con el uso de un modo ALWIP. Evitar la necesidad de almacenar tales datos puede reducir el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0105] En algunos ejemplos, se puede inferir que el valor de un `alwip_flag` de una fila

de CTU anterior es igual a un valor particular (por ejemplo, 0) con el propósito de derivar el contexto de `alwip_flag` para el bloque actual. En otras palabras, el codificador de video 200 y el decodificador de video 300 pueden asumir que los valores de `alwip_flag` para los bloques vecinos anteriores que están en una fila de CTU anterior tienen todos el mismo valor particular. De esta manera, el codificador de video 200 y el decodificador de video 300 pueden evitar la necesidad de almacenar datos que indiquen si los bloques en la fila de CTU por encima de la fila de CTU que contiene el bloque actual están codificados con el uso de un modo ALWIP, con las consiguientes reducciones potenciales de costo y complejidad para el codificador de video 200 y el decodificador de video 300.

[0106] Por lo tanto, de acuerdo con un ejemplo de la primera técnica de esta divulgación, un codificador de video (por ejemplo, un codificador de video 200 o un decodificador de video 300) puede determinar un contexto para un elemento de sintaxis ALWIP (por ejemplo, `alwip_flag`). En este ejemplo, el elemento de sintaxis ALWIP indica si un bloque actual está codificado con ALWIP. Además, en este ejemplo, se determina que un bloque vecino de arriba no está disponible para determinar el contexto en función de que el bloque vecino de arriba está en una CTU diferente del bloque actual, siendo el bloque vecino de arriba un vecino de arriba del bloque actual. En este ejemplo, el codificador de video puede codificar el elemento de sintaxis ALWIP en función del contexto. Por ejemplo, el codificador de video puede usar el contexto en la codificación CABAC del elemento de sintaxis ALWIP.

[0107] De acuerdo con una segunda técnica de esta divulgación, se introduce una restricción para inhabilitar ALWIP para bloques menores que un umbral particular de ancho, w_T y altura, h_T . En otras palabras, un estándar de codificación de video (por ejemplo, VVC, etc.) puede imponer una restricción que deshabilita ALWIP para bloques que tienen un ancho menor que un ancho de umbral o una altura menor que una altura de umbral. Por lo tanto, un estándar de codificación de video puede especificar que los bloques que tienen anchos menores que w_T y alturas menores que h_T no se pueden codificar con el uso de ALWIP. Por ejemplo, la restricción puede proporcionar que ALWIP esté deshabilitado para bloques con anchos menores que 8 y alturas menores que 8. En algunos ejemplos, los elementos de sintaxis relacionados con ALWIP están condicionados en función de tales restricciones de tamaño de bloque. Como se señaló con anterioridad, por ejemplo, con respecto a la Figura 12, la codificación de bloques pequeños con ALWIP puede resultar en altos requisitos de almacenamiento, pero ganancias relativamente modestas en la eficiencia de codificación. La introducción de

esta restricción puede reducir la cantidad de parámetros que el codificador de video 200 y el decodificador de video 300 pueden necesitar almacenar para implementar ALWIP, reduciendo así potencialmente la complejidad y el costo del codificador de video 200 y el decodificador de video 300. Por lo tanto, de acuerdo con la segunda técnica de la presente divulgación, se puede imponer una restricción que deshabilite ALWIP para bloques que tienen anchos menores que un ancho de umbral (por ejemplo, 8) o alturas menores que una altura de umbral (por ejemplo, 8).

[0108] De acuerdo con una tercera técnica de esta divulgación, cuando un bloque vecino codificado con ALWIP se considera/prueba en la derivación MPM de un bloque codificado no ALWIP, el bloque vecino puede considerarse no disponible (o intercodificado) y/o se puede inferir que el valor del modo intra de tal bloque vecino es un valor particular (por ejemplo, modo planar, modo DC u otro modo de intrapredicción) para la derivación MPM del bloque codificado no ALWIP. Para facilitar la explicación, esta divulgación puede referirse al valor particular como valor predeterminado. De esta manera, el codificador de video 200 y el decodificador de video 300 pueden evitar la necesidad de almacenar una tabla de mapeo para mapear los modos ALWIP a modos de intrapredicción no ALWIP y pueden evitar operaciones de búsqueda en la tabla de mapeo. Esto puede reducir los requisitos de almacenamiento del codificador de video 200 y el decodificador de video 300, lo que puede reducir el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0109] Por lo tanto, de acuerdo con la tercera técnica de esta divulgación, un codificador de video (por ejemplo, codificador de video 200 o decodificador de video 300) puede realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual que no está codificado con el uso de ALWIP. Como parte de la realización del proceso de derivación de MPM, el codificador de video puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. En función de que el bloque vecino es un bloque vecino codificado con ALWIP, el codificador de video puede determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado. El codificador de video puede incluir entonces un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM. Si el bloque vecino no está codificado con ALWIP, el codificador de video puede determinar que el MPM o los MPM incluyen el modo de intrapredicción del bloque vecino. El codificador de video puede codificar (por ejemplo, codificar o decodificar) el bloque actual en función de uno de los MPM para el bloque actual o en función de otro modo de intrapredicción.

[0110] Además, en algunos ejemplos, cuando un bloque de croma se codifica con el modo DM (es decir, el modo de intrapredicción del bloque de croma se hereda de un bloque de luma contiguo) y el bloque de luma contiguo (o más particularmente, un bloque de luma usado para derivar el modo DM del bloque de croma) está codificado con ALWIP, el modo intra del bloque contiguo puede determinarse, por ejemplo, inferirse, como un valor predeterminado (por ejemplo, planar). De esta manera, el codificador de video 200 y el decodificador de video 300 pueden evitar la necesidad de almacenar una tabla de mapeo para mapear los modos ALWIP a modos de intrapredicción no ALWIP y pueden evitar operaciones de búsqueda en la tabla de mapeo. Esto puede reducir los requisitos de almacenamiento del codificador de video 200 y el decodificador de video 300, lo que puede reducir el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0111] Por lo tanto, en algunos ejemplos en los que un bloque de croma se codifica con un modo DM, un codificador de video (por ejemplo, un codificador de video 200 o un decodificador de video 300) puede determinar si un bloque de luma contiguo está codificado con ALWIP, donde el bloque de luma contiguo se coloca con el bloque de croma. En función del bloque de luma contiguo que se codifica con ALWIP, el codificador de video puede determinar, por ejemplo, inferir, que un modo de intrapredicción del bloque de luma contiguo es un valor predeterminado (por ejemplo, modo planar), como el modo planar u otro modo de intrapredicción. El codificador de video puede codificar (por ejemplo, codificar o decodificar) el bloque de croma con el uso de un modo de intrapredicción correspondiente al valor predeterminado. Por ejemplo, para codificar el bloque de croma con el uso del modo de intrapredicción correspondiente al valor predeterminado, el codificador de video 200 puede generar un bloque de predicción con el uso del modo de intrapredicción. El codificador de video 200 puede generar entonces datos residuales para el bloque cromático en función de muestras del bloque de croma y el bloque de predicción. Para decodificar el bloque de croma con el uso del modo de intrapredicción correspondiente al valor predeterminado, el decodificador de video 300 puede generar un bloque de predicción con el uso del modo de intrapredicción. El decodificador de video 300 puede reconstruir entonces el bloque de croma en función del bloque de predicción y los datos residuales del bloque de croma.

[0112] De acuerdo con una cuarta técnica de esta divulgación, cuando se usa un bloque vecino no ALWIP en la derivación MPM de un bloque ALWIP, el bloque vecino puede considerarse no disponible (o intercodificado) y/o el valor del modo vecino puede asignarse a un valor predeterminado para la derivación MPM del bloque ALWIP. En

otras palabras, al determinar ALWIP-MPM para un bloque codificado con ALWIP y un bloque vecino no está codificado con el uso de ALWIP, el codificador de video 200 y el decodificador de video 300 pueden considerar que el bloque vecino no está disponible o pueden asumir que el bloque vecino tiene un modo de intrapredicción predeterminado. Esto puede evitar o reducir la necesidad de que el codificador de video 200 y el decodificador de video 300 almacenen datos de mapeo para convertir modos de intrapredicción direccional no ALWIP en modos ALWIP. Evitar o reducir la necesidad de almacenar tales datos de mapeo puede reducir el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0113] En algunos ejemplos, si un bloque vecino se codifica con el modo intra/inter combinado (CIIP), el bloque vecino se puede considerar como disponible y se le puede asignar el modo planar, u otro modo no ALWIP apropiado, para la derivación MPM del bloque ALWIP. Alternativamente, el modo CIIP puede mapearse directamente a un modo ALWIP apropiado para optimizar los beneficios de eficiencia de codificación. La asignación puede depender de las dimensiones del bloque, etc. Si el bloque vecino está codificado con el modo de copia intrabloque (IBC), el bloque se puede considerar como disponible y se le puede asignar el modo planar, u otro modo apropiado que no sea ALWIP, para la derivación MPM del bloque ALWIP. Alternativamente, el modo IBC puede mapearse directamente a un modo ALWIP apropiado para optimizar los beneficios de eficiencia de codificación. La asignación puede depender de las dimensiones del bloque, etc. El mapeo de los modos CIIP e IBC directamente (por ejemplo, en una relación 1 a 1) a un modo particular, como el modo planar, puede evitar o reducir la necesidad de almacenar datos de mapeo más complejos. Evitar o reducir la necesidad de almacenar datos de mapeo más complejos puede reducir el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0114] De acuerdo con una quinta técnica de esta divulgación, se puede introducir una restricción para inhabilitar ALWIP para bloques con una relación de ancho a altura de 4 o 0,25. En otras palabras, un estándar de codificación de video puede proporcionar que un bloque que tenga una relación de ancho a altura de 4 o 0,25 no pueda codificarse con el uso de ALWIP. No codificar bloques que tengan tales relaciones de ancho a altura puede reducir el número de parámetros ALWIP que el codificador de video 200 y el decodificador de video 300 pueden necesitar almacenar, reduciendo así potencialmente el costo y la complejidad del codificador de video 200 y el decodificador de video 300.

[0115] La derivación de la lista del MPM para ALWIP usa los modos de los bloques vecinos que están adyacentes a la parte superior derecha (por ejemplo, justo encima de

la muestra superior derecha) y/o muestras de la parte inferior izquierda (por ejemplo, justo a la izquierda de la muestra de la parte inferior izquierda) del bloque actual. En algunos ejemplos, la ubicación vecina usada para derivar los MPM para ALWIP está alineada con la ubicación vecina usada para derivar las MPM para la intrapredicción regular. Más generalmente, uno o más bloques en la vecindad del bloque actual pueden usarse para la derivación de MPM para ALWIP. La Figura 13 es un diagrama conceptual que ilustra ejemplos de bloques vecinos 550, 552 usados para derivar ALWIP MPM para un bloque actual 554, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 13, el bloque A (550) es un bloque (superior) adyacente a la muestra superior derecha del bloque actual 554. El bloque L (552) es un bloque (izquierdo) adyacente a la muestra inferior izquierda del bloque actual 554.

[0116] El siguiente texto de especificación de JVET-N0217 se edita para implementar las técnicas descritas con respecto a la cuarta técnica de esta divulgación. Los cambios se indican mediante etiquetas `<!--...-->`.

Asignación de modo al bloque codificado con CIIP para la derivación MPM del bloque ALWIP

Proceso de derivación para el modo de intrapredicción ponderada lineal afín

Las entradas a este proceso son:

- una ubicación de luma (x_{Cb} , y_{Cb}) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior izquierda de la imagen actual,
- una variable $cbWidth$ que especifica el ancho del bloque de codificación actual en muestras de luma,
- una variable $cbHeight$ que especifica la altura del bloque de codificación actual en muestras de luma.

En este proceso, se deriva el modo de intrapredicción ponderada lineal afín $IntraPredModeY[x_{Cb}][y_{Cb}]$.

$IntraPredModeY[x_{Cb}][y_{Cb}]$ se deriva mediante los siguientes pasos ordenados:

1. Las ubicaciones vecinas (x_{NbA} , y_{NbA}) y (x_{NbB} , y_{NbB}) se establecen iguales a ($x_{Cb} - 1$, y_{Cb}) y (x_{Cb} , $y_{Cb} - 1$), respectivamente.
 2. Para que X sea reemplazado por A o B, las variables $candLwipModeX$ se derivan de la siguiente manera:
- El proceso de derivación de disponibilidad para un bloque como se especifica en la cláusula 6.4.X [Ed. (BB): proceso de verificación de disponibilidad de bloques

vecinos tbd] se invoca con la ubicación (xCurr, yCurr) establecida igual a (xCb, yCb) y la ubicación vecina (xNbY, yNbY) establecida igual a (xNbX, yNbX) como entradas, y la salida se asigna a availableX.

- El modo de intrapredicción ponderada lineal afín candidato candLwipModeX se obtiene de la siguiente manera:
 - Si una o más de las siguientes condiciones son verdaderas, candLwipModeX se establece igual a -1.
 - La variable availableX es igual a FALSE.
 - CuPredMode [xNbX] [yNbX] no es igual a MODE_INTRA y ciip_flag[xNbX][yNbX] no es igual a 1.
 - pcm_flag[xNbX][yNbX] es igual a 1.
 - X es igual a B e yCb - 1 es menor que ((yCb >> CtbLog2SizeY) << CtbLog2SizeY).
 - De lo contrario, se aplica lo siguiente:
 - El proceso de derivación del tipo de tamaño para un bloque como se especifica en la cláusula 8.4.X.1 se invoca con el ancho del bloque de codificación actual en muestras de luma cbWidth y la altura del bloque de codificación actual en muestras de luma cbHeight como entrada, y la salida se asigna a la variable sizeId.
 - Si intra_lwip_flag[xNbX][yNbX] es igual a 1, el proceso de derivación del tipo de tamaño para un bloque como se especifica en la cláusula 8.4.X.1 se invoca con el ancho del bloque de codificación vecino en las muestras luma nbWidthX y la altura del bloque de codificación vecino en muestras luma nbHeightX como entrada, y la salida se asigna a la variable sizeIdX.
 - Si sizeId es igual a sizeIdX, candLwipModeX se establece igual a IntraPredModeY[xNbX][yNbX].
 - De lo contrario, candLwipModeX se establece igual a -1.
 - <!-- De lo contrario, si intra_lwip_flag[xNbX][yNbX] es igual a 0 y ciip_flag ciip_flag[xNbX][yNbX] es igual a 1, IntraPredModeY[xNbX][yNbX] es igual a INTRA_PLANAR y candLwipModeX se deriva con el uso de IntraPredModeY[xNbX][yNbX] y sizeId como se especifica en la Tabla 8-X1.-->
 - De lo contrario, candLwipModeX se deriva con el uso de

IntraPredModeY[xNbX][yNbX] y sizeId como se especifica en la Tabla 8-X1.

3. CandLwipModeList [x] con x = 0..2 se deriva de la siguiente manera, con el uso de lwipMpmCand [sizeId] como se especifica en la Tabla 8-X2:

- Si candLwipModeA y candLwipModeB son ambos iguales a -1, se aplica lo siguiente:

$$\text{candLwipModeList}[0] = \text{lwipMpmCand}[\text{sizeId}][0] \quad (8-X1)$$

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8-X2)$$

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X3)$$

- De lo contrario, se aplica lo siguiente:

- Si candLwipModeA es igual a candLwipModeB o si candLwipModeA o candLwipModeB es igual a -1, se aplica lo siguiente:

$$\begin{aligned} \text{candLwipModeList}[0] &= \\ &(\text{candLwipModeA} \neq -1) ? \text{candLwipModeA} : \text{candLwipModeB} \quad (8-X4) \end{aligned}$$

- Si candLwipModeList [0] es igual a lwipMpmCand [sizeId] [0], se aplica lo siguiente:

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8-X5)$$

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X6)$$

- De lo contrario, se aplica lo siguiente:

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][0] \quad (8-X7)$$

$$\begin{aligned} \text{candLwipModeList}[2] &= \\ &(\text{candLwipModeList}[0] \neq \text{lwipMpmCand}[\text{sizeId}][1]) ? \\ &\text{lwipMpmCand}[\text{sizeId}][1] : \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X8) \end{aligned}$$

- De lo contrario, se aplica lo siguiente:

$$\text{candLwipModeList}[0] = \text{candLwipModeA} \quad (8-X9)$$

$$\text{candLwipModeList}[1] = \text{candLwipModeB} \quad (8-X10)$$

- Si candLwipModeA y candLwipModeB no son iguales a lwipMpmC y [sizeId] [0], se aplica lo siguiente:

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][0] \quad (8-X11)$$

- De lo contrario, se aplica lo siguiente:

- Si `candLwipModeA` y `candLwipModeB` no son iguales a `lwipMpmC` y `[sizeId][1]`, se aplica lo siguiente:

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8\text{-X12})$$
 - De lo contrario, se aplica lo siguiente:

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8\text{-X13})$$
4. `IntraPredModeY[xCb][yCb]` se deriva al aplicar el siguiente procedimiento:
- Si `intra_lwip_mpm_flag[xCb][yCb]` es igual a 1, `IntraPredModeY[xCb][yCb]` se establece igual a `candLwipModeList[intra_lwip_mpm_idx[xCb][yCb]]`.
 - De lo contrario, `IntraPredModeY[xCb][yCb]` se deriva al aplicar los siguientes pasos ordenados:
 1. Cuando `candLwipModeList[i]` es mayor que `candLwipModeList[j]` para $i = 0..1$ y para cada $i, j = (i + 1)..2$, ambos valores se intercambian de la siguiente manera:

$$\begin{aligned} &(\text{candLwipModeList}[i], \quad \text{candLwipModeList}[j]) = \\ &\text{Swap}(\text{candLwipModeList}[i], \quad \text{candLwipModeList}[j]) \end{aligned} \quad (8\text{-X14})$$
 2. `IntraPredModeY[xCb][yCb]` se deriva mediante los siguientes pasos ordenados:
 - i. `IntraPredModeY[xCb][yCb]` se establece igual a `intra_lwip_mpm_remainder[xCb][yCb]`.
 - ii. Para i igual a 0 a 2, inclusive, cuando `IntraPredModeY[xCb][yCb]` es mayor o igual que `candLwipModeList[i]`, el valor de `IntraPredModeY[xCb][yCb]` se incrementa en uno.

La variable `IntraPredModeY[x][y]` con $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ e $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$ se establece para que sea igual a `IntraPredModeY[xCb][yCb]`.

8.4.X.1 Proceso de derivación para el tipo de tamaño de bloque de predicción

Las entradas a este proceso son:

- una variable `cbWidth` que especifica el ancho del bloque de codificación actual en muestras de luma,

- una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma.

La salida de este proceso es una variable sizeld.

La variable sizeld se deriva de la siguiente manera:

- Si tanto cbWidth como cbHeight son iguales a 4, sizeld se establece igual a 0.
- De lo contrario, si tanto cbWidth como cbHeight son menores o iguales a 8, sizeld se establece igual a 1.
- De lo contrario, sizeld se establece igual a 2.

Tabla 8-X1 - Especificación del mapeo entre los modos de intrapredicción y de intrapredicción ponderada lineal afín

IntraPredModeY[xNbX][yNbX]	tipo de tamaño de bloque sizeld		
	0	1	2
0	17	0	5
1	17	0	1
2, 3	17	10	3
4, 5	9	10	3
6, 7	9	10	3
8, 9	9	10	3
10, 11	9	10	0
12, 13	17	4	0
14, 15	17	6	0
16, 17	17	7	4
18, 19	17	7	4
20, 21	17	7	4
22, 23	17	5	5
24, 25	17	5	1
26, 27	5	0	1
28, 29	5	0	1
30, 31	5	3	1
32, 33	5	3	1
34, 35	34	12	6
36, 37	22	12	6
38, 39	22	12	6
40, 41	22	12	6
42, 43	22	14	6
44, 45	34	14	10
46, 47	34	14	10
48, 49	34	16	9
50, 51	34	16	9
52, 53	34	16	9
54, 55	34	15	9
56, 57	34	13	9
58, 59	26	1	8
60, 61	26	1	8
62, 63	26	1	8
64, 65	26	1	8

IntraPredModeY[xNbX][yNbX]	tipo de tamaño de bloque sized		
	0	1	2
66	26	1	8

Tabla 8-X2 - Especificación de modos candidatos de intrapredicción ponderada lineal afín

	modo candidato		
	0	1	2
lwipMpmCand[0]	17	34	5
lwipMpmCand[1]	0	7	16
lwipMpmCand[2]	1	4	6

[0117] Los cambios de ejemplo en el texto de especificación de JVET-N0217 para implementar la quinta técnica de esta divulgación se indican con los indicadores <!-->... </!> a continuación.

Proceso de derivación para el modo de intrapredicción ponderada lineal afín

Las entradas en este proceso son:

- una ubicación de luma (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra luma superior izquierda de la imagen actual,
- una variable cbWidth que especifica el ancho del bloque de codificación actual en muestras de luma,
- una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma.

En este proceso, se deriva el modo de intrapredicción ponderada lineal afín IntraPredModeY[xCb][yCb].

IntraPredModeY[xCb][yCb] se deriva mediante los siguientes pasos ordenados:

1. Las ubicaciones vecinas (xNbA, yNbA) y (xNbB, yNbB) se establecen igual a (xCb – 1, yCb <!-->+ cbHeight – 1</!>) y (xCb <!-->+ cbWidth – 1</!>, yCb – 1), respectivamente.
2. Para que X sea reemplazado por A o B, las variables candLwipModeX se derivan de la siguiente manera:
 - El proceso de derivación de disponibilidad para un bloque como se especifica en la cláusula 6.4.X [Ed. (BB): proceso de verificación de disponibilidad de bloques vecinos tbd] se invoca con la ubicación (xCurr, yCurr) establecida igual a (xCb, yCb) y la ubicación vecina (xNbY, yNbY) establecida igual a (xNbX, yNbX)

como entradas, y la salida se asigna a availableX.

- El modo de intrapredicción ponderada lineal afín candidato candLwipModeX se obtiene de la siguiente manera:
 - Si una o más de las siguientes condiciones son verdaderas, candLwipModeX se establece igual a -1.
 - La variable availableX es igual a FALSE.
 - CuPredMode[xNbX][yNbX] no es igual a MODE_INTRA y ciip_flag[xNbX][yNbX] no es igual a 1.
 - pcm_flag [xNbX] [yNbX] es igual a 1.
 - X es igual a B e yCb - 1 es menor que ((yCb >> CtbLog2SizeY) << CtbLog2SizeY).
 - De lo contrario, se aplica lo siguiente:
 - El proceso de derivación del tipo de tamaño para un bloque como se especifica en la cláusula 8.4.X.1 se invoca con el ancho del bloque de codificación actual en muestras de luma cbWidth y la altura del bloque de codificación actual en muestras de luma cbHeight como entrada, y la salida se asigna a la variable sizeld.
 - Si intra_lwip_flag [xNbX] [yNbX] es igual a 1, el proceso de derivación del tipo de tamaño para un bloque como se especifica en la cláusula 8.4.X.1 se invoca con el ancho del bloque de codificación vecino en las muestras de luma nbWidthX y la altura del bloque de codificación vecino en muestras de luma nbHeightX como entrada, y la salida se asigna a la variable sizeldX.
 - Si sizeld es igual a sizeldX, candLwipModeX se establece igual a IntraPredModeY[xNbX][yNbX].
 - De lo contrario, candLwipModeX se establece igual a -1.
 - De lo contrario, candLwipModeX se deriva con el uso de IntraPredModeY[xNbX][yNbX] y sizeld

como se especifica en la Tabla 8-X1.

3. CandLwipModeList [x] con $x = 0..2$ se deriva de la siguiente manera, con el uso de lwipMpmCand [sizeId] como se especifica en la Tabla 8-X2:

- Si candLwipModeA y candLwipModeB son ambos iguales a -1 , se aplica lo siguiente:

$$\text{candLwipModeList}[0] = \text{lwipMpmCand}[\text{sizeId}][0] \quad (8-X1)$$

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8-X2)$$

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X3)$$

- De lo contrario, se aplica lo siguiente:

- Si candLwipModeA es igual a candLwipModeB o si candLwipModeA o candLwipModeB es igual a -1 , se aplica lo siguiente:

$$\begin{aligned} \text{candLwipModeList}[0] &= \\ &(\text{candLwipModeA} \neq -1) ? \text{candLwipModeA} : \text{candLwipModeB} \quad (8-X4) \end{aligned}$$

- Si candLwipModeList [0] es igual a lwipMpmCand [sizeId] [0], se aplica lo siguiente:

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8-X5)$$

$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X6)$$

- De lo contrario, se aplica lo siguiente:

$$\text{candLwipModeList}[1] = \text{lwipMpmCand}[\text{sizeId}][0] \quad (8-X7)$$

$$\begin{aligned} \text{candLwipModeList}[2] &= \\ &(\text{candLwipModeList}[0] \neq \text{lwipMpmCand}[\text{sizeId}][1]) ? \end{aligned}$$

$$\text{lwipMpmCand}[\text{sizeId}][1] : \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X8)$$

- De lo contrario, se aplica lo siguiente:

$$\text{candLwipModeList}[0] = \text{candLwipModeA} \quad (8-X9)$$

$$\text{candLwipModeList}[1] = \text{candLwipModeB} \quad (8-X10)$$

- Si candLwipModeA y candLwipModeB no son iguales a lwipMpmC y [sizeId] [0], se aplica lo

siguiente:

$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][0]$
(8-X11)

- De lo contrario, se aplica lo siguiente:
 - Si candLwipModeA y candLwipModeB no son iguales a lwipMpmC y $[\text{sizeId}][1]$, se aplica lo siguiente:
$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][1] \quad (8-X12)$$
 - De lo contrario, se aplica lo siguiente:
$$\text{candLwipModeList}[2] = \text{lwipMpmCand}[\text{sizeId}][2] \quad (8-X13)$$

4. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se deriva al aplicar el siguiente procedimiento:

- Si $\text{intra\_lwip\_mpm\_flag}[\text{xCb}][\text{yCb}]$ es igual a 1, $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se establece igual a $\text{candLwipModeList}[\text{intra\_lwip\_mpm\_idx}[\text{xCb}][\text{yCb}]]$.
- De lo contrario, $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se deriva al aplicar los siguientes pasos ordenados:
 1. Cuando $\text{candLwipModeList}[i]$ es mayor que $\text{candLwipModeList}[j]$ para $i = 0..1$ y para cada $i, j = (i + 1)..$
2, ambos valores se intercambian de la siguiente manera:
$$(\text{candLwipModeList}[i], \text{candLwipModeList}[j]) = \text{Swap}(\text{candLwipModeList}[i], \text{candLwipModeList}[j]) \quad (8-X14)$$
 2. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se deriva mediante los siguientes pasos ordenados:
 - i. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se establece igual a $\text{intra\_lwip\_mpm\_remainder}[\text{xCb}][\text{yCb}]$.
 - ii. Para i igual a 0 a 2, inclusive, cuando $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ es mayor o igual que $\text{candLwipModeList}[i]$, el valor de $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se incrementa en uno.

La variable $\text{IntraPredModeY}[x][y]$ con $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ e $y = \text{yCb}.. \text{yCb}$

+ cbHeight - 1 se establece para que sea igual a IntraPredModeY[xCb][yCb].

Tabla 8-X1 - Especificación del mapeo entre los modos de intrapredicción y de intrapredicción ponderada lineal afín

IntraPredModeY[xNbX][yNbX]	tipo de tamaño de bloque sizeId		
	0	1	2
0	17	0	5
1	17	0	1
2, 3	17	10	3
4, 5	9	10	3
6, 7	9	10	3
8, 9	9	10	3
10, 11	9	10	0
12, 13	17	4	0
14, 15	17	6	0
16, 17	17	7	4
18, 19	17	7	4
20, 21	17	7	4
22, 23	17	5	5
24, 25	17	5	1
26, 27	5	0	1
28, 29	5	0	1
30, 31	5	3	1
32, 33	5	3	1
34, 35	34	12	6
36, 37	22	12	6
38, 39	22	12	6
40, 41	22	12	6
42, 43	22	14	6
44, 45	34	14	10
46, 47	34	14	10
48, 49	34	16	9
50, 51	34	16	9
52, 53	34	16	9
54, 55	34	15	9
56, 57	34	13	9
58, 59	26	1	8
60, 61	26	1	8
62, 63	26	1	8
64, 65	26	1	8
66	26	1	8

Tabla 8-X2 - Especificación de modos candidatos de intrapredicción ponderada lineal afín

	modo candidato		
	0	1	2
lwipMpmCand[0]	17	34	5
lwipMpmCand[1]	0	7	16
lwipMpmCand[2]	1	4	6

[0118] Esta divulgación generalmente puede referirse a “señalizar” cierta información, tal como elementos de sintaxis. El término “señalización” puede referirse generalmente a la comunicación de valores de elementos de sintaxis y/u otros datos utilizados para decodificar datos de video codificados. Es decir, el codificador de video 200 puede señalar valores para elementos de sintaxis en el flujo de bits. En general, la señalización se refiere a generar un valor en el flujo de bits. Como se indicó con anterioridad, el dispositivo de origen 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir al almacenar elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

[0119] Las Figuras 14A y 14B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario quadtree (QTBT) 130, y una unidad de árbol de codificación correspondiente (CTU) 132. Las líneas continuas representan la división 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 sub-bloques con el mismo tamaño. En consecuencia, el codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar elementos de sintaxis (tal como la información de división) para un nivel de árbol de región (es decir, el primer nivel) de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (tal como la información de división) para un nivel de árbol de predicción (es decir, el segundo nivel) de la estructura QTBT 130 (es decir, las líneas discontinuas). El codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar datos de video, tales como datos de predicción y transformación, para CU representadas por nodos hoja terminales de la estructura QTBT 130.

[0120] En general, CTU 132 de la Figura 14B puede estar asociado con parámetros que definen tamaños de bloques correspondientes a nodos de la estructura QTBT 130 en el primer y segundo nivel. Estos parámetros pueden incluir un tamaño de CTU (que representa un tamaño de CTU 132 en muestras), un tamaño mínimo de quadtree (MinQTSIZE, que representa un tamaño mínimo permitido de nodo hoja de quadtree), un tamaño máximo del árbol binario (MaxBTSIZE, que representa un tamaño máximo permitido de nodo raíz del árbol binario), una profundidad máxima del árbol binario

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

[0121] 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 tales nodos con la inclusión del nodo primario y los nodos secundarios con líneas continuas para las ramas. Si los nodos del primer nivel no son mayores que el tamaño máximo permitido del nodo raíz del árbol binario (MaxBTSize), pueden dividirse aún más por los respectivos árboles binarios. La división del árbol binario de un nodo puede repetirse hasta que los nodos resultantes de la división alcancen el tamaño mínimo permitido del nodo hoja del árbol binario (MinBTSize) o la profundidad máxima permitida del árbol binario (MaxBTDepth). El ejemplo de la estructura QTBT 130 representa tales nodos con líneas discontinuas para las ramas. El nodo hoja de árbol binario se conoce como una unidad de codificación (CU), que se utiliza para la predicción (por ejemplo, predicción intraimagen o interimagen) y se transforma, sin ninguna partición más. Como se discutió con anterioridad, las CU también pueden denominarse “bloques de video” o “bloques”.

[0122] En un ejemplo de la estructura de partición QTBT, el tamaño de la CTU se establece en 128x128 (muestras de luma y dos muestras de croma de 64x64 correspondientes), el MinQTSIZE se establece en 16x16, el MaxBTSize se establece en 64x64, el MinBTSize (para ancho y altura) se establece en 4, y MaxBTDepth se establece en 4. La partición quadtree se aplica primero a la CTU para generar nodos hoja de quadtree. Los nodos hoja de quadtree pueden tener un tamaño de 16x16 (es decir, el MinQTSIZE) a 128x128 (es decir, el tamaño de la CTU). Si el nodo hoja de quadtree es de 128x128, no se dividirá más por el árbol binario, ya que el tamaño excede el MaxBTSize (es decir, 64x64, en este ejemplo). De lo contrario, el árbol binario dividirá más el nodo hoja de quadtree. Por lo tanto, el nodo hoja de quadtree es también el nodo raíz del árbol binario y tiene una profundidad del árbol binario como 0. Cuando la profundidad del árbol binario alcanza MaxBTDepth (4, en este ejemplo), no se permite más división. Cuando el nodo del árbol binario tiene un ancho igual a MinBTSize (4, en este ejemplo), implica que no se permite más división vertical. De manera similar, un nodo de árbol binario que tiene una altura igual a MinBTSize implica que no se permite más división horizontal para ese nodo de árbol binario. Como se señaló con anterioridad,

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

[0123] La Figura 15 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede realizar las técnicas de la presente divulgación. La Figura 15 se proporciona con fines explicativos y no debe considerarse una limitación de las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 en el contexto de estándares de codificación de video tales como el estándar de codificación de video HEVC y el estándar de codificación de video H.266 en desarrollo. Sin embargo, las técnicas de la presente divulgación no se limitan a estos estándares de codificación de video y son aplicables generalmente al encriptado y decodificación de video.

[0124] En el ejemplo de la Figura 15, el codificador de video 200 incluye la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el búfer de imagen decodificada (DPB) 218 y unidad de encriptado de entropía 220. Cualquiera o todos de la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el DPB 218 y la unidad de encriptado de entropía 220 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el codificador de video 200 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

[0125] 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 (Figura 1). El DPB 218 puede actuar como una memoria de imagen de referencia que almacena datos de video de referencia para su uso en la predicción de datos de video posteriores por el codificador de video 200. La memoria de datos de video 230 y el DPB 218 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), que incluye DRAM síncrona (SDRAM), RAM magnetorresistiva

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

[0126] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describa específicamente como tal, o la memoria externa al codificador de video 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como memoria de referencia que almacena datos de video que el encriptador de video 200 recibe para encriptar (por ejemplo, datos de video para un bloque actual que se encriptará). La memoria 106 de la Figura 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

[0127] Las diversas unidades de la Figura 15 se ilustran para ayudar a comprender las operaciones realizadas por el codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de estos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0128] El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU), unidades de funciones elementales (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En los ejemplos en los que las operaciones del codificador de video 200 se realizan con el uso de un *software* ejecutado por los circuitos programables, la memoria 106 (Figura 1) puede almacenar el código objeto del *software* que el codificador de video 200 recibe y ejecuta,

u otra memoria dentro del codificador de video 200 (no mostrada) puede almacenar tales instrucciones.

[0129] 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.

[0130] 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. Por ejemplo, en el ejemplo de la Figura 15, la unidad de intrapredicción 226 incluye una unidad ALWIP 227 configurada para realizar ALWIP.

[0131] La unidad de selección de modo 202 generalmente coordina múltiples pases de encriptado para probar combinaciones de parámetros de encriptado y los valores de distorsión de velocidad resultantes para tales combinaciones. Los parámetros de encriptado pueden incluir la división de CTU en CU, modos de predicción para las CU, tipos de transformación para datos residuales de las CU, parámetros de cuantificación para datos residuales de las CU, etc. La unidad de selección de modo 202 puede finalmente seleccionar la combinación de parámetros de encriptado que tienen valores de distorsión de velocidad que son mejores que las otras combinaciones probadas.

[0132] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 230 en una serie de CTU, y encapsular una o más CTU dentro de un segmento. La unidad de selección de modo 202 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura QTBT o la estructura quadtree de HEVC descrita con anterioridad. Como se describió con anterioridad, el codificador de video 200 puede formar una o más CU de particionar una CTU de acuerdo con la estructura de árbol. Tal CU también se puede denominar generalmente como un “bloque de video” o “bloque”.

[0133] En general, la unidad de selección de modo 202 también controla sus

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

[0134] 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 con el uso de los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia con el uso del vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por los respectivos vectores de movimiento y combinar los datos recuperados, por ejemplo, mediante el promedio de muestra por muestra o el

promedio ponderado.

[0135] 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.

[0136] De acuerdo con un ejemplo de la tercera técnica de esta divulgación, la unidad de intrapredicción 226 puede realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de los datos de video. En este ejemplo, el bloque actual no se codifica mediante ALWIP. Como parte de la realización del proceso de derivación de MPM, la unidad de intrapredicción 226 puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. En función de que el bloque vecino es un bloque vecino codificado con ALWIP, la unidad de intrapredicción 226 puede determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. La unidad de intrapredicción 226 puede incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM. El codificador de video 200 puede codificar el bloque actual en función de uno de los MPM para el bloque actual u otro modo de intrapredicción. Por ejemplo, como parte del encriptador de video 200 que encripta el bloque actual, la unidad de intrapredicción 226 puede usar uno de los MPM u otro modo de intrapredicción para generar un bloque de predicción para el bloque actual.

[0137] 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 y sin procesar 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 con el uso de la

modulación de código de pulso diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse con el uso de uno o más circuitos sustractores que realizan sustracción binaria.

[0138] En los ejemplos en los que la unidad de selección de modo 202 divide las CU en PU, cada PU puede asociarse con una PU de luma y las PU de croma correspondientes. El codificador de video 200 y el decodificador de video 300 pueden admitir PU que tienen varios tamaños. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es 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.

[0139] En los ejemplos en los que la unidad de selección de modo no divide más una CU en PU, cada CU puede estar asociada con un bloque de codificación de luma y los bloques de codificación croma correspondientes. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU. El codificador de video 200 y el decodificador de video 300 pueden admitir tamaños de CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

[0140] Para otras técnicas de codificación de video tales como una codificación en modo de copia intrabloque, una codificación en modo afín y una codificación en modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de las unidades respectivas asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se codifica. En algunos ejemplos, como la codificación de modo de paleta, la unidad de selección de modo 202 puede no generar un bloque de predicción, y en su lugar generar elementos de sintaxis que indican la manera de reconstruir el bloque en función de una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de encriptado de entropía 220 a encriptar.

[0141] Como se describió con anterioridad, la unidad de generación residual 204 recibe los datos de video para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 genera entonces un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque de predicción y el bloque actual.

[0142] La unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en este documento “bloque de coeficiente de transformación”). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional, una transformada de Karhunen-Loeve (KLT) o una transformada conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformación 206 puede aplicar múltiples transformadas a un bloque residual, por ejemplo, una transformada primaria y una transformada secundaria, tal como una transformada rotacional. En algunos ejemplos, la unidad de procesamiento de transformación 206 no aplica transformadas a un bloque residual.

[0143] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficiente de transformación, para producir un bloque de coeficiente de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación de un bloque de coeficiente de transformación de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, a través de la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con el bloque actual al ajustar el valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información y, por lo tanto, los coeficientes de transformación cuantificados pueden tener una precisión menor que los coeficientes de transformación originales producidos por la unidad de procesamiento de transformación 206.

[0144] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformación inversa 212 pueden aplicar cuantificación inversa y transformaciones inversas a un bloque de coeficiente de transformación cuantificado, respectivamente, para reconstruir un bloque residual del bloque de coeficiente de transformación. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con cierto grado de distorsión) en función del 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.

[0145] 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.

[0146] El codificador de video 200 almacena bloques reconstruidos en el DPB 218. Por ejemplo, en los ejemplos donde no se realizan operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en el DPB 314. En ejemplos donde se realizan operaciones de la unidad de filtro 312, la unidad de filtro 312 puede almacenar los bloques reconstruidos filtrados en el DPB 314. Por ejemplo, en ejemplos donde no se necesitan operaciones de la unidad de filtro 216, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en el DPB 218. En ejemplos donde se necesitan operaciones de la unidad de filtro 216, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados en el DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia del DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para interpredicir los bloques de imágenes codificadas posteriormente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en el DPB 218 de una imagen actual para intrapredicir otros bloques en la imagen actual.

[0147] En general, la unidad de encriptado de entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de encriptado de entropía 220 puede codificar por entropía bloques de coeficientes de transformación cuantificados de la unidad de cuantificación 208. Como otro ejemplo, la unidad de encriptado de entropía 220 puede codificar por entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para interpredicción o información intramodal para intrapredicción) desde la unidad de selección de modo 202. La unidad de encriptado de entropía 220 puede realizar una o más operaciones de encriptado de entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de encriptado de entropía 220 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de Entropía de Partición de Intervalo de Probabilidad (PIPE),

una operación de encriptado exponencial-Golomb u otro tipo de operación de encriptado de entropía en los datos. En algunos ejemplos, la unidad de encriptado de entropía 220 puede funcionar en modo de derivación donde los elementos de sintaxis no están encriptados por entropía.

[0148] El codificador de video 200 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o imagen. En particular, la unidad de encriptado de entropía 220 puede emitir el flujo de bits.

[0149] Las operaciones descritas con anterioridad se describen con respecto a un bloque. Dicha descripción debe entenderse como operaciones para un bloque de codificación de luma y/o bloques de codificación de croma. Como se describió con anterioridad, en algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una CU. En algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una PU.

[0150] En algunos ejemplos, las operaciones realizadas con respecto a un bloque de codificación de luma no necesitan repetirse para los bloques de codificación de croma. Como un ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación de luma no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de croma. Por el contrario, el MV para el bloque de codificación de luma se puede escalar para determinar el MV para los bloques de croma, y la imagen de referencia puede ser la misma. Como otro ejemplo, el proceso de intrapredicción puede ser el mismo para los bloques de codificación de luma y los bloques de codificación de croma.

[0151] 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 ALWIP de esta divulgación. Por ejemplo, el codificador de video 200 puede representar un ejemplo de un dispositivo que incluye una memoria configurada para almacenar datos de video y una o más unidades de procesamiento implementadas en circuitos y configuradas para realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de los datos de video, en donde el bloque actual no se codifica mediante ALWIP. En este ejemplo, el procesador o los procesadores se configuran de manera que, como parte de la realización del proceso de derivación de MPM, el procesador o los procesadores

determinan si un bloque vecino del bloque actual es un bloque vecino codificado en ALWIP. Además, el procesador o los procesadores están configurados para, en función de que el bloque vecino es un bloque vecino codificado con ALWIP, determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. El procesador o los procesadores pueden configurarse además para incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM. El procesador o los procesadores pueden codificar el bloque actual en función de uno de los MPM para el bloque actual.

[0152] Además, en algunos ejemplos donde un bloque es un bloque de croma codificado con un modo DM, el codificador de video 200 representa un ejemplo de un dispositivo que incluye una memoria configurada para almacenar datos de video y uno o más procesadores configurados para determinar si un bloque de luma contiguo está codificado con ALWIP. En tales ejemplos, el bloque de luma contiguo se coloca junto con el bloque. Además, en tales ejemplos, el procesador o los procesadores pueden, en función del bloque de luma contiguo que se codifica con ALWIP, determinar, por ejemplo, inferir, que un modo de intrapredicción del bloque de luma contiguo es un valor predeterminado. En otras palabras, el procesador o los procesadores pueden determinar o tratar de otro modo el modo de intrapredicción del bloque de luma contiguo como si fuera un modo de intrapredicción predeterminado, tal como el modo de intrapredicción planar. El procesador o los procesadores pueden configurarse para codificar el bloque con el uso de un modo de intrapredicción correspondiente al valor predeterminado.

[0153] La Figura 16 es un diagrama de bloques que ilustra un decodificador de video 300 de ejemplo que puede realizar las técnicas de la presente divulgación. La Figura 16 se proporciona con fines explicativos y no se limita a las técnicas que se ejemplifican y describen ampliamente en esta divulgación. A efectos de explicación, esta divulgación describe el decodificador 300 de video según las técnicas de JEM, VVC y HEVC. Sin embargo, las técnicas de la presente divulgación pueden realizarse mediante dispositivos de codificación de video que están configurados para otros estándares de codificación de video.

[0154] En el ejemplo de la Figura 16, el decodificador de video 300 incluye la memoria de búfer de imagen codificada (CPB) 320, la unidad de decodificación de entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el búfer de imagen decodificada (DPB) 314. Cualquiera o

todos de la memoria de CPB 320, la unidad de decodificación de entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el DPB 314 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el decodificador de video 300 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

[0155] 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. Por ejemplo, en el ejemplo de la Figura 16, la unidad de intrapredicción 318 incluye una unidad ALWIP 319 que realiza un proceso ALWIP. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o diferentes componentes funcionales.

[0156] 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 (Figura 1). La memoria de CPB 320 puede incluir un CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. Además, la memoria 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. El 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 el DPB 314 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales 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 el DPB 314 pueden ser proporcionados 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.

[0157] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (Figura 1). Es decir, la memoria 120 puede almacenar datos como se discutió con anterioridad con la memoria de CPB 320. Del mismo modo, la memoria 120 puede almacenar instrucciones para ser ejecutadas por el decodificador de video 300, cuando parte o la totalidad de la funcionalidad del decodificador de video 300 se implementa en el *software* para ser ejecutado por circuitos de procesamiento del decodificador de video 300.

[0158] Las diversas unidades mostradas en la Figura 16 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 Figura 15, 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, una o más de las unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0159] 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.

[0160] La unidad de decodificación de entropía 302 puede recibir datos de video codificados del CPB y decodificar por entropía los datos de video para reproducir elementos de sintaxis. La unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308,

la unidad de reconstrucción 310 y la unidad de filtro 312 pueden generar datos de video decodificados basados en los elementos de sintaxis extraídos del flujo de bits.

[0161] En general, el decodificador de video 300 reconstruye una imagen bloque por bloque. El decodificador de video 300 puede realizar una operación de reconstrucción en cada bloque individualmente (donde el bloque que se está reconstruyendo actualmente, es decir, decodificado, puede denominarse un “bloque actual”).

[0162] La unidad de decodificación de entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de coeficiente de transformación cuantificado, así como información de transformación, tal como un parámetro de cuantificación (QP) y/o indicaciones de modo de transformación. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficiente de transformación cuantificado para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que aplique la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda bit a bit para cuantificar inversamente los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede formar así un bloque de coeficiente de transformación que incluye coeficientes de transformación.

[0163] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficiente de transformación para generar un bloque residual asociado con el bloque actual. Por ejemplo, la unidad de procesamiento de transformación inversa 308 puede aplicar una DCT inversa, una transformada entera inversa, una transformada de Karhunen-Loeve inversa (KLT), una transformada rotacional inversa, una transformada direccional inversa u otra transformación inversa al bloque de coeficientes de transformación.

[0164] Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con los elementos de sintaxis de información de predicción que fueron decodificados por entropía por la unidad de decodificación de entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es interpredicho, la unidad de compensación de movimiento 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de información de predicción pueden indicar una imagen de referencia en el DPB 314 desde la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con la

ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede realizar el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de compensación de movimiento 224 (Figura 15).

[0165] Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es intrapredicho, la unidad de intrapredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. Nuevamente, la unidad de intrapredicción 318 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 (Figura 15). La unidad de interpredicción 318 puede recuperar datos de muestras vecinas al bloque actual del DPB 314.

[0166] De acuerdo con un ejemplo de la tercera técnica de esta divulgación, la unidad de intrapredicción 318 puede realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de los datos de video. En este ejemplo, el bloque actual no se codifica mediante ALWIP. Como parte de la realización del proceso de derivación de MPM, la unidad de intrapredicción 318 puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. En función de que el bloque vecino es un bloque vecino codificado con ALWIP, la unidad de intrapredicción 318 puede determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. La unidad de intrapredicción 318 puede incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM. El decodificador de video 300 puede decodificar el bloque actual en función de uno de los MPM para el bloque actual u otro modo de intrapredicción. Por ejemplo, como parte del decodificador de video 300 que codifica el bloque actual, la unidad de intrapredicción 318 puede usar uno de los MPM u otro modo de intrapredicción para generar un bloque de predicción para el bloque actual.

[0167] La unidad de reconstrucción 310 puede reconstruir el bloque actual con el uso del bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0168] 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.

[0169] El decodificador de video 300 puede almacenar los bloques reconstruidos en el DPB 314. Como se discutió con anterioridad, el DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes decodificadas previamente para la compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas desde el DPB para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la Figura 1.

[0170] 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 ALWIP de esta divulgación. Por ejemplo, el decodificador de video 300 puede representar un ejemplo de un dispositivo que incluye una memoria configurada para almacenar datos de video y una o más unidades de procesamiento implementadas en circuitos y configuradas para realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de datos de video, en donde el bloque actual no se codifica mediante ALWIP. En este ejemplo, el procesador o los procesadores se configuran de manera que, como parte de la realización del proceso de derivación de MPM, el procesador o los procesadores determinan si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP. Además, el procesador o los procesadores están configurados para, en función de que el bloque vecino es un bloque vecino codificado con ALWIP, determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, tal como un valor correspondiente a un modo planar. El procesador o los procesadores pueden incluir un modo de intrapredicción correspondiente al predeterminado como uno del MPM o los MPM. El procesador o los procesadores pueden decodificar el bloque actual en función de uno de los MPM para el bloque actual.

[0171] Además, en algunos ejemplos donde un bloque es un bloque de croma codificado con un modo DM, el codificador de video 200 representa un ejemplo de un dispositivo que incluye una memoria configurada para almacenar datos de video y uno o más procesadores configurados para determinar si un bloque de luma contiguo está codificado con ALWIP. En tales ejemplos, el bloque de luma contiguo se coloca junto con el bloque. Además, en tales ejemplos, el procesador o los procesadores pueden, en

función del bloque de luma contiguo que se codifica con ALWIP, determinar, por ejemplo, inferir, que un modo de intrapredicción del bloque de luma contiguo es un valor predeterminado. En otras palabras, el procesador o los procesadores pueden determinar o tratar de otro modo el modo de intrapredicción del bloque de luma contiguo como si fuera un modo de intrapredicción predeterminado, tal como el modo de intrapredicción planar. El procesador o los procesadores se pueden configurar para decodificar el bloque con el uso de un modo de intrapredicción correspondiente al valor predeterminado.

[0172] La Figura 17 es un diagrama de flujo que ilustra un método de ejemplo para encriptar un bloque actual. El bloque actual puede incluir una CU actual. Aunque se describe con respecto al encriptador de video 200 (Figuras 1 y 15), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la Figura 17.

[0173] 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. En algunos ejemplos, el codificador de video 200 (por ejemplo, la unidad de intrapredicción 226 del codificador de video 200) realiza un proceso de derivación de MPM que deriva uno o más MPM para el bloque actual. En este ejemplo, el bloque actual no se codifica mediante ALWIP. Los MPM pueden incluir modos de intrapredicción y, en algunos ejemplos, cada uno de los MPM es un modo de intrapredicción. Además, en este ejemplo, como parte de la realización del proceso de derivación de MPM, el codificador de video 200 puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP y, en función de que el bloque vecino es un bloque vecino codificado con ALWIP, determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado. El codificador de video 200 puede incluir un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM. El codificador de video 200 puede entonces usar uno de los MPM u otro modo de intrapredicción para formar el bloque de predicción para el bloque actual.

[0174] 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 transformar y cuantificar entonces los coeficientes de transformación del bloque residual (354).

[0175] A continuación, el codificador de video 200 puede explorar los coeficientes de

transformación cuantificados del bloque residual (356). Durante la exploración, o después de la exploración, el codificador de video 200 puede codificar por entropía los coeficientes de transformación (358). Por ejemplo, el codificador de video 200 puede codificar los coeficientes de transformación con el uso de CAVLC o CABAC. El codificador de video 200 puede entonces emitir los datos codificados por entropía del bloque (360).

[0176] La Figura 18 es un diagrama de flujo que ilustra un método 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 (Figuras 1 y 16), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la Figura 18.

[0177] 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 transformación de un bloque residual correspondiente al bloque actual (370). El decodificador de video 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y reproducir los coeficientes de transformación del bloque residual (372).

[0178] El decodificador de video 300 puede predecir el bloque actual (374), por ejemplo, con el uso de un modo de intrapredicción o interpredicción como lo indica la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual. En algunos ejemplos, el decodificador de video 300 (por ejemplo, la unidad de intrapredicción 318 del decodificador de video 300) realiza un proceso de derivación de MPM que deriva uno o más MPM para el bloque actual. En este ejemplo, el bloque actual no se codifica mediante ALWIP. Los MPM pueden incluir modos de intrapredicción y, en algunos ejemplos, cada uno de los MPM es un modo de intrapredicción. Además, en este ejemplo, como parte de la realización del proceso de derivación de MPM, el decodificador de video 300 puede determinar si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP y, en función de que el bloque vecino es un bloque vecino codificado con ALWIP, determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado, como un valor correspondiente a un modo planar. El decodificador de video 300 puede incluir un modo de intrapredicción correspondiente al valor predeterminado (por ejemplo, el modo planar) como uno del MPM o los MPM. El decodificador de video 300 puede entonces usar uno de los MPM u otro modo de intrapredicción para formar el bloque de predicción

para el bloque actual.

[0179] El decodificador de video 300 puede entonces explorar inversamente los coeficientes de transformación reproducidos (376), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de video 300 puede entonces cuantificar inversamente los coeficientes de transformación y aplicar una o más transformaciones inversas a los coeficientes de transformación para producir un bloque residual (378). El decodificador de video 300 puede decodificar el bloque actual al combinar el bloque de predicción y el bloque residual (380).

[0180] La Figura 19 es un diagrama de flujo que ilustra un método de ejemplo de codificación de datos de video, de acuerdo con una o más técnicas de esta divulgación. El método de la Figura 19 puede ser realizado por un codificador de video, tal como un codificador de video 200 o un decodificador de video 300.

[0181] En el ejemplo de la Figura 19, el codificador de video puede realizar un proceso de derivación de MPM que deriva uno o más MPM para un bloque actual de los datos de video (400). En algunos ejemplos, la unidad de intrapredicción 226 (Figura 15) del codificador 200 de video realiza el proceso de derivación de MPM. En algunos ejemplos, la unidad de intrapredicción 318 (Figura 16) del decodificador de video 300 realiza el proceso de derivación de MPM. En el ejemplo de la Figura 19, el bloque actual no se codifica mediante ALWIP. Los MPM pueden incluir modos de intrapredicción y, en algunos ejemplos, cada uno de los MPM es un modo de intrapredicción. En los ejemplos en los que el codificador de video es un codificador de video (por ejemplo, el codificador de video 200), el codificador de video puede estar incluido en un dispositivo (por ejemplo, el dispositivo de origen 102) que incluye una cámara que captura una imagen que incluye el bloque actual.

[0182] Además, en el ejemplo de la Figura 19, realizar el proceso de derivación de MPM puede incluir que el codificador de video determine si un bloque vecino del bloque actual es un bloque vecino codificado con ALWIP (402). El bloque vecino puede ser un bloque vecino de arriba o un bloque vecino izquierdo del bloque actual. En algunos ejemplos, el codificador de video puede almacenar `alwip_flags` para bloques de la imagen actual y puede usar las `alwip_flags` almacenadas del bloque vecino para determinar si el bloque vecino está codificado con el uso de ALWIP.

[0183] En función de que el bloque vecino es un bloque vecino codificado con ALWIP, el codificador de video puede determinar que un valor de un modo de intrapredicción del bloque vecino es un valor predeterminado (404). Por ejemplo, el codificador de video puede determinar (por ejemplo, inferir) el valor del modo intra del bloque vecino como

un valor particular que indica el modo planar. En otras palabras, el modo de intrapredicción correspondiente al valor predeterminado puede ser el modo planar. En otros ejemplos, el modo de intrapredicción correspondiente al valor predeterminado puede ser otro modo de intrapredicción. El codificador de video puede incluir entonces un modo de intrapredicción correspondiente al valor predeterminado como uno del MPM o los MPM para el bloque actual (406).

[0184] El codificador de video puede codificar el bloque actual en función de uno de los MPM para el bloque actual (408). Por ejemplo, en los ejemplos en los que el codificador de video es un codificador de video como el codificador de video 200, el codificador de video puede codificar el bloque actual en función de uno de los MPM para el bloque actual. En tales ejemplos, el codificador de video puede usar el MPM para generar un bloque de predicción para el bloque actual. El codificador de video puede entonces usar el bloque de predicción y muestras del bloque actual para generar datos residuales para el bloque actual. El codificador de video puede entonces procesar los datos residuales para el bloque actual como se describe en otra parte de esta divulgación, por ejemplo, como con respecto a la Figura 15. En los ejemplos en los que el codificador de video es un decodificador de video como el decodificador de video 300, el decodificador de video puede decodificar el bloque actual en función de uno de los MPM para el bloque actual. En tales ejemplos, el decodificador de video puede usar el MPM para generar un bloque de predicción para el bloque actual. El decodificador de video puede entonces usar el bloque de predicción y los datos residuales del bloque actual para reconstruir el bloque actual. Además, en algunos ejemplos, un dispositivo que incluye el decodificador de video puede incluir una pantalla que muestra una imagen que incluye el bloque actual.

[0185] La Figura 20 es un diagrama de flujo que ilustra un método de ejemplo de codificación de un bloque de croma, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 20, un codificador de video (por ejemplo, un codificador de video 200 o un decodificador de video 300) puede codificar un bloque de croma con el uso de un modo DM.

[0186] En particular, el codificador de video puede determinar si un bloque de luma contiguo está codificado con ALWIP (450). El bloque de luma contiguo se coloca con el bloque de croma. En función del bloque de luma contiguo que se codifica con ALWIP, el codificador de video puede determinar que un valor de un modo de intrapredicción del bloque de luma contiguo es un valor predeterminado, tal como un valor correspondiente al modo planar.

[0187] El codificador de video puede entonces codificar el bloque de croma con el uso

de un modo de intrapredicción correspondiente al valor predeterminado (454). Por ejemplo, en los ejemplos en los que el codificador de video es un codificador de video como el codificador de video 200, el codificador de video puede generar un bloque de predicción para el bloque de croma con el uso del modo de intrapredicción correspondiente al valor predeterminado. El codificador de video puede entonces usar el bloque de predicción y muestras del bloque de croma para generar datos residuales para el bloque de croma. El codificador de video puede entonces procesar los datos residuales para el bloque de croma como se describe en otra parte de esta divulgación, por ejemplo, como con respecto a la Figura 15. En los ejemplos en los que el codificador de video es un decodificador de video como el decodificador de video 300, el decodificador de video puede decodificar el bloque de croma en función del modo de intrapredicción correspondiente al valor predeterminado. En tales ejemplos, el decodificador de video puede usar el modo de intrapredicción correspondiente al valor predeterminado para generar un bloque de predicción para el bloque de croma. El decodificador de video puede entonces usar el bloque de predicción y los datos residuales del bloque de croma para reconstruir el bloque de croma. Además, en algunos ejemplos, un dispositivo que incluye el decodificador de video puede incluir una pantalla que muestra una imagen que incluye el bloque de croma.

[0188] Los siguientes párrafos proporcionan un conjunto no limitativo de ejemplos que están de acuerdo con una o más técnicas de esta divulgación.

[0189] Ejemplo 1. Un método de codificación de datos de video, que incluye: determinar un contexto para un elemento de sintaxis de intrapredicción ponderada lineal afín (ALWIP) que indica si un bloque actual está codificado con ALWIP, en donde se determina que un bloque vecino de arriba no está disponible para determinar el contexto en función de que el bloque vecino de arriba está en una unidad de árbol de codificación (CTU) diferente del bloque actual, siendo el bloque vecino de arriba un vecino de arriba del bloque actual; y codificar el elemento de sintaxis ALWIP en función del contexto.

[0190] Ejemplo 2. Un método de codificación de datos de video, que incluye: codificar un conjunto de bloques de datos de video, en donde se impone una restricción en la codificación que deshabilita la intrapredicción ponderada lineal afín (ALWIP) para aquellos de los bloques que tienen un ancho menor que un ancho de umbral o una altura menor que una altura de umbral.

[0191] Ejemplo 3. Un método de codificación de datos de video, que incluye: determinar que un bloque vecino para un bloque actual de los datos de video se considera no disponible en función de que el bloque vecino está codificado por intrapredicción

ponderada lineal afín (ALWIP) y el bloque vecino se considera en una derivación del modo más probable (MPM) de un bloque codificado no ALWIP de los datos de video; determinar MPM para el bloque actual en función de bloques vecinos disponibles; y codificar el bloque actual en función de uno de los MPM.

[0192] Ejemplo 4. Un método de codificación de datos de video, que incluye: asignar un valor predeterminado para una derivación del modo más probable (MPM) de un bloque ALWIP en función de la determinación de que un bloque vecino para un bloque actual de los datos de video se considera no disponible en función de que el bloque vecino no está codificado con intrapredicción ponderada lineal afín (ALWIP) y el bloque vecino se usa en una derivación del modo más probable (MPM) de un bloque de los datos de video codificado con ALWIP; y usar ALWIP para codificar el bloque actual.

[0193] Ejemplo 5. Un método de codificación de datos de video, que incluye: derivar una lista del modo más probable (MPM) para la intrapredicción ponderada lineal afín (ALWIP) en función de uno o más bloques en una vecindad de un bloque actual de los datos de video; y usar un MPM de la lista del MPM para realizar ALWIP para codificar el bloque actual.

[0194] Ejemplo 6. El método de acuerdo con el ejemplo 5, en donde el bloque o los bloques en la vecindad del bloque actual incluyen un primer bloque vecino que está encima y adyacente a una muestra superior derecha del bloque actual y un segundo bloque vecino que está a la izquierda y adyacente a una muestra de la parte inferior izquierda del bloque actual.

[0195] Ejemplo 7. Un método que incluye los métodos de acuerdo con cualquier combinación de uno o más de los ejemplos 1-6.

[0196] Ejemplo 8. El método de acuerdo con cualquiera de los ejemplos 1-7, en donde la codificación incluye la decodificación.

[0197] Ejemplo 9. El método de acuerdo con cualquiera de los ejemplos 1-7, en donde la codificación incluye el encriptado.

[0198] Ejemplo 10. Un dispositivo para codificar datos de video, en donde el dispositivo incluye uno o más medios para realizar el método de acuerdo con cualquiera de los ejemplos 1-9.

[0199] Ejemplo 11. El dispositivo de acuerdo con el ejemplo 10, en donde el medio o los medios incluyen uno o más procesadores implementados en circuitos.

[0200] Ejemplo 12. El dispositivo de acuerdo con cualquiera de los ejemplos 10 y 11, que incluye además una memoria para almacenar los datos de video.

[0201] Ejemplo 13. El dispositivo de acuerdo con cualquiera de los ejemplos 10-12, que

incluye además una pantalla configurada para mostrar datos de video decodificados.

[0202] Ejemplo 14. El dispositivo de acuerdo con cualquiera de los ejemplos 10-13, en donde el dispositivo incluye uno o más de una cámara, una computadora, un dispositivo móvil, un dispositivo receptor de transmisión o un decodificador.

[0203] Ejemplo 15. El dispositivo de acuerdo con cualquiera de los ejemplos 10-14, en donde el dispositivo incluye un decodificador de video.

[0204] Ejemplo 16. El dispositivo de acuerdo con cualquiera de los ejemplos 10-15, en donde el dispositivo incluye un codificador de video.

[0205] Ejemplo 17. Un medio de almacenamiento legible por computadora que tiene almacenadas instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen el método de acuerdo con cualquiera de los ejemplos 1-7.

[0206] Ejemplo 18. Un dispositivo para codificar datos de video, donde el dispositivo incluye medios para realizar los métodos de acuerdo con cualquiera de los ejemplos 1-7.

[0207] Debe reconocerse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos descritos o eventos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través de procesamiento de subprocesos múltiples, procesamiento de interrupción o procesadores múltiples, en lugar de secuencialmente.

[0208] En uno o más ejemplos, las funciones descritas pueden implementarse en *hardware*, *software*, *firmware* o cualquier combinación de estos. Si se implementan en *software*, las funciones pueden almacenarse o transmitirse como una o más instrucciones o códigos en un medio legible por computadora y ejecutarse por una unidad de procesamiento basada en *hardware*. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, que corresponden a un medio tangible como los medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa de computadora de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios tangibles de almacenamiento legibles por computadora que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder una o más computadoras o uno o más

procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa de computadora puede incluir un medio legible por computadora.

[0209] A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por computadora pueden incluir RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que puede usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que puede acceder una computadora. Además, cualquier conexión se denomina correctamente un medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota con el uso de un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por computadora y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en este documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computadora.

[0210] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señal digital (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices de compuerta programables en campo (FPGA) u otro equivalente integrado o circuitos lógicos discretos. En consecuencia, los términos “procesador” y “circuito de procesamiento”, como se usan en el presente documento, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos dedicados de *hardware* y/o *software* configurados para encriptado y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en

uno o más circuitos o elementos lógicos.

[0211] Las técnicas de la presente divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de *hardware*. Por el contrario, como se describió con anterioridad, varias unidades pueden combinarse en una unidad de *hardware* de códec o proporcionarse mediante una colección de unidades de *hardware* interoperativas, que incluyen uno o más procesadores como se describe con anterioridad, junto con el *software* y/o *firmware* adecuado.

[0212] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.