

RECONSTRUCCIONES DE VECTORES DE MOVIMIENTO PARA FLUJO ÓPTICO BIDIRECCIONAL (BIO)

[0001] La presente solicitud reivindica el beneficio de:

la solicitud de patente provisional N.º 62/442.357 presentada el 4 de enero de 2017; y

la solicitud de patente provisional N.º 62/445.152 presentada el 11 de enero de 2017,

cuyo contenido se incorpora en la presente por referencia.

CAMPO TÉCNICO

[0002] La presente descripción se refiere a la codificación de video.

ANTECEDENTES

[0003] Las capacidades de video digital se pueden incorporar a un amplio rango de dispositivos, que incluyen televisores digitales, sistemas digitales de transmisión directa, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, tablets, e-readers, cámaras digitales, dispositivos de registro digitales, transmisores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radioteléfonos celulares o satelitales, denominados “smartphones”, dispositivos de videoteleconferencia, dispositivos de streaming de videos, y similares. Los dispositivos de video digital implementan técnicas de codificación de video, tales como aquellas 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 las extensiones de estos estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficaz implementando las técnicas de codificación de video.

[0004] Las técnicas de codificación de video incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia inherente en secuencias de video. Para la codificación de video en bloques, un segmento de video (por ejemplo, un cuadro de video o una porción de un cuadro de video) se puede particionar en bloques de video, que también se pueden denominar bloques de árbol (*treeblocks*), unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en un segmento intracodificado (I) de una imagen se pueden codificar usando la predicción espacial con respecto a muestras de

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

[0005] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se desea codificar. Los datos residuales representan las diferencias de píxeles entre el bloque original que se desea codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta hacia un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar del dominio de píxeles a un dominio de la transformada, lo que da como resultado coeficientes de la transformada residuales, que luego se pueden cuantificar. Los coeficientes de la transformada cuantificados, inicialmente dispuestos en una matriz de dos dimensiones, se pueden escanear para producir un vector unidimensional de coeficientes de la transformada, y se puede aplicar codificación entrópica para alcanzar una compresión aún mayor.

SÍNTESIS

[0006] En general, la presente divulgación describe técnicas relacionadas con el flujo óptico bidireccional (BIO) en la codificación de video. Las técnicas de la presente divulgación se pueden usar junto con los códecs de video existentes, tales como codificación de video de alta eficiencia (HEVC), o pueden ser una herramienta de codificación eficaz para estándares de codificación de video futuros.

[0007] De acuerdo con un ejemplo de la presente divulgación, un método para decodificar datos de video incluye determinar un bloque de datos de video que se codifica mediante un modo de interpredicción bidireccional; determinar un primer vector de movimiento (MV) para el bloque, en donde el primer MV apunta a una primera imagen de referencia; determinar un segundo MV para el bloque, en donde el segundo MV apunta a una segunda imagen de referencia, en donde la primera imagen de referencia es diferente a la segunda imagen de referencia; mediante el primer MV, localizar un primer bloque predictivo en la primera imagen de referencia; mediante el segundo MV, localizar un segundo bloque predictivo en la segunda imagen de

referencia; para un primer subbloque del primer bloque predictivo, determinar una primera cantidad de movimiento de flujo óptico bidireccional (BIO); determinar un primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, un primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO; para un segundo subbloque del primer bloque predictivo, determinar una segunda cantidad de movimiento de BIO; determinar un segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO; en función del primer subbloque predictivo final y el segundo subbloque predictivo final, determinar un bloque predictivo final para el bloque de datos de video; y proporcionar una imagen de datos de video que comprende una versión decodificada del bloque de datos de video.

[0008] De acuerdo con otro ejemplo de la presente divulgación, un dispositivo para decodificar datos de video incluye una memoria configurada para almacenar datos de video; y uno o más procesadores configurados para determinar un bloque de datos de video que se codifica mediante un modo de interpredicción bidireccional; determinar un primer vector de movimiento (MV) para el bloque, en donde el primer MV apunta a una primera imagen de referencia; determinar un segundo MV para el bloque, en donde el segundo MV apunta a una segunda imagen de referencia, en donde la primera imagen de referencia es diferente a la segunda imagen de referencia; mediante el primer MV, localizar un primer bloque predictivo en la primera imagen de referencia; mediante el segundo MV, localizar un segundo bloque predictivo en la segunda imagen de referencia; para un primer subbloque del primer bloque predictivo, determinar una primera cantidad de movimiento de flujo óptico bidireccional (BIO); determinar un primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, un primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO; para un segundo subbloque del primer bloque predictivo, determinar una segunda cantidad de movimiento de BIO; determinar un segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO; en función del primer subbloque predictivo final y el segundo subbloque predictivo final, determinar un bloque predictivo final para el bloque de datos de video; y proporcionar una imagen de datos de video que comprende una

versión decodificada del bloque de datos de video.

[0009] De acuerdo con otro ejemplo de la presente divulgación, un medio de almacenamiento legible por computadora almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que el uno o más procesadores determinen un bloque de datos de video que se codifica mediante un modo de interpredicción bidireccional; determinen un primer vector de movimiento (MV) para el bloque, en donde los primeros el primer MV apunta a una primera imagen de referencia; determinen un segundo MV para el bloque, en donde el segundo MV apunta a una segunda imagen de referencia, en donde la primera imagen de referencia es diferente a la segunda imagen de referencia; mediante el primer MV, localicen un primer bloque predictivo en la primera imagen de referencia; mediante el segundo MV, localicen un segundo bloque predictivo en la segunda imagen de referencia; para un primer subbloque del primer bloque predictivo, determinen una primera cantidad de movimiento de flujo óptico bidireccional (BIO); determinen un primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, un primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO; para un segundo subbloque del primer bloque predictivo, determinen una segunda cantidad de movimiento de BIO; determinen un segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO; en función del primer subbloque predictivo final y el segundo subbloque predictivo final, determinen un bloque predictivo final para el bloque de datos de video; y proporcionen una imagen de datos de video que comprende una versión decodificada del bloque de datos de video.

[0010] De acuerdo con otro ejemplo de la presente divulgación, un aparato para decodificar datos de video incluye un medio para determinar un bloque de datos de video que se codifica mediante un modo de interpredicción bidireccional; un medio para determinar un primer vector de movimiento (MV) para el bloque, en donde el primer MV apunta a una primera imagen de referencia; un medio para determinar un segundo MV para el bloque, en donde el segundo MV apunta a una segunda imagen de referencia, en donde la primera imagen de referencia es diferente a la segunda imagen de referencia; un medio para localizar un primer bloque predictivo en la primera imagen de referencia; mediante el primer MV; un medio para localizar un segundo bloque predictivo en la segunda imagen de referencia mediante el segundo

MV; un medio para determinar una primera cantidad de movimiento de flujo óptico bidireccional (BIO) para un primer subbloque del primer bloque predictivo; un medio para determinar un primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, un primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO; un medio para determinar una segunda cantidad de movimiento de BIO para un segundo subbloque del primer bloque predictivo; un medio para determinar un segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO; un medio para determinar un primer bloque predictivo para el bloque de datos de video en función del primer subbloque predictivo final y el segundo subbloque predictivo final; y un medio para proporcionar una imagen de datos de video que comprende una versión decodificada del bloque de datos de video.

[0011] Los detalles de uno o más aspectos de la presente divulgación se indican en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas de las técnicas descritas en la presente divulgación serán evidentes de la descripción, dibujos y reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0012] La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video que puede utilizar técnicas para flujo óptico bidireccional.

[0013] La Figura 2 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) unilateral como un algoritmo de comparación de bloques (BMA) realizado para la conversión ascendente de la velocidad de cuadro de movimiento compensado (MC-FRUC).

[0014] La Figura 3 es un diagrama conceptual que ilustra un ejemplo de ME bilateral como un BMA realizado para la MC-FRUC.

[0015] La Figura 4A muestra MV candidatos vecinos espaciales para el modo de fusión.

[0016] La Figura 4B muestra MV candidatos vecinos espaciales para los modos de AMVP.

[0017] La Figura 5A muestra un ejemplo de un TMVP candidato.

[0018] La Figura 5B muestra un ejemplo de escalamiento de MV.

[0019] La Figura 6 muestra un ejemplo de trayectoria de flujo óptico.

- [0020]** La Figura 7 muestra un ejemplo de BIO para un bloque 8x4.
- [0021]** La Figura 8 muestra un ejemplo de BIO modificado para un bloque 8x4.
- [0022]** Las Figuras 9A y 9B muestran ilustraciones de ejemplo de subbloques en donde se aplica OBMC.
- [0023]** Las Figuras 10A-10D muestra ejemplo de ponderaciones de OBMC.
- [0024]** La Figura 11 muestra un ejemplo para el BIO propuesto para un bloque 8x4.
- [0025]** Las Figuras 12A-12D muestran ejemplos del BIO simplificado propuesto en la OBMC.
- [0026]** La Figura 13 un ejemplo de la función de ponderación para un subbloque 4x4 con una ventana 5x5.
- [0027]** La Figura 14 es un diagrama en bloque que ilustra un ejemplo de un codificador de video.
- [0028]** La Figura 15 es un diagrama de bloques que ilustra un ejemplo de decodificador de video que puede implementar técnicas para flujo óptico bidireccional.
- [0029]** La Figura 16 es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de video, de acuerdo con una técnica de la presente divulgación.

DESCRIPCIÓN DETALLADA

[0030] En general, las técnicas de la presente divulgación se refieren a mejoras de técnicas de codificación de video de flujo óptico bidireccional (BIO). El BIO se puede aplicar durante la compensación de movimiento. Como se sugirió originalmente, el BIO se usa para modificar valores de muestra predictivos para bloques intercodificados previsto bidireccionalmente en función de una trayectoria de flujo óptico con el fin de determinar mejores bloques predictivos, por ejemplo, bloques predictivos que coinciden en mayor medida con un bloque original de datos de video. Las distintas técnicas de divulgación se pueden aplicar, solas o combinadas, para determinar si llevar a cabo BIO, y cuándo, al predecir bloques de datos de video, por ejemplo, durante la compensación de movimiento.

[0031] Como se usa en la presente divulgación, la expresión codificación de video en general se refiere a codificación de video o decodificación de video. De manera similar, la expresión codificador de video se puede referir en general a un codificador de video o a un decodificador de video. Asimismo, determinadas técnicas descritas en la divulgación con respecto a la decodificación de video también se pueden aplicar a la codificación de video, y viceversa. Por ejemplo, a menudo los codificadores de video y los decodificadores de video se configuran para llevar a cabo el mismo proceso o procesos recíprocos. Además, los codificadores de video generalmente realizan

decodificación de video como parte de los procesos para determinar cómo codificar datos de video. Por lo tanto, a menos que se exprese explícitamente lo contrario, no se debería asumir que una técnica descrita con respecto a la decodificación de video no se puede realizar también mediante un codificador de video, o viceversa.

[0032] La presente divulgación también puede usar expresiones como capa actual, bloque actual, imagen actual, segmento actual, etc. En el contexto de la presente divulgación, mediante el término actual se pretende identificar un bloque, imagen, segmento, etc. que se está codificando actualmente, en contraposición con, por ejemplo, bloques, imágenes o segmentos ya codificados o codificados previamente, o bloques, imágenes o segmentos que aún no se codificaron.

[0033] En general, una imagen se divide en bloques, cada uno de los cuales se puede codificar de manera predictiva. Un codificador de video puede predecir un bloque actual mediante técnicas de intrapredicción (usando datos de la imagen que incluye el bloque actual), técnicas de interpredicción (usando datos de una imagen previamente codificada en relación con la imagen que incluye el bloque actual), u otras técnicas tales como copia intrabloque, modo paleta, modo diccionario, etc. La interpredicción incluye tanto predicción unidireccional como predicción bidireccional.

[0034] Para cada bloque interprevisto, un codificador de video puede determinar un conjunto de información de movimiento. El conjunto de información de movimiento puede contener información de movimiento para direcciones de predicción en sentido directo e indirecto. En la presente, las direcciones de predicción en sentido directo e indirecto son dos direcciones de predicción de un modo de predicción bidireccional. Los términos “directo” e “indirecto” no necesariamente tienen un significado geométrico. En su lugar, los términos corresponden por lo general a si las imágenes de referencia se van a mostrar antes (“directo”) o después (“indirecto”) de la imagen actual. En algunos ejemplos, las direcciones de predicción “directa” e “indirecta” pueden corresponder a la lista de imágenes de referencia 0 (RefPicList0) y a la lista de imágenes de referencia 1 (RefPicList1) de una imagen actual. Cuando solo una lista de imágenes de referencia está disponible para una imagen o un segmento, solo RefPicList0 se encuentra disponible, y la información de movimiento de cada bloque de un segmento siempre se refiere a una imagen de RefPicList0 (por ejemplo, es directa).

[0035] En algunos casos, se puede usar un vector de movimiento junto con un índice de referencia correspondiente en un proceso de decodificación. Tal vector de movimiento con un índice de referencia asociado se indica como un conjunto

unipredictivo de información de movimiento.

[0036] Para cada dirección de predicción, la información de movimiento contiene un índice de referencia y un vector de movimiento. En algunos casos, a los fines de la simplicidad, se puede hacer referencia a un vector de movimiento en sí mismo de manera que se asume que el vector de movimiento tiene un índice de referencia asociado. Se puede usar un índice de referencia para identificar una imagen de referencia en la lista de imágenes de referencia actual (RefPicList0 o RefPicList1). Un vector de movimiento tiene un componente horizontal (x) y uno vertical (y). En general, el componente horizontal indica un desplazamiento horizontal dentro de una imagen de referencia, en relación con la posición de un bloque actual en una imagen actual, necesaria para localizar una coordenada x de un bloque de referencia, mientras que el componente vertical indica un desplazamiento vertical dentro de la imagen de referencia, en relación con la posición del bloque actual, necesaria para localizar una coordenada y del bloque de referencia.

[0037] Los valores del contador de orden de imágenes (POC) se usan ampliamente en los estándares de codificación de video para identificar un orden de visualización de una imagen. Si bien existen casos en los que dos imágenes dentro de una secuencia de video codificada pueden tener el mismo valor de POC, en general, esto no ocurre dentro de una secuencia de video codificada. De este modo, los valores de POC de las imágenes en general son únicos, y así pueden identificar de manera única las imágenes correspondientes. Cuando múltiples secuencias de video codificadas se encuentran presentes en una corriente de bits, las imágenes que tienen un mismo valor de POC pueden estar más cerca entre sí en cuanto al orden de decodificación. En general, los valores de POC de las imágenes se usan para construcción de listas de imágenes de referencia, derivación de conjuntos de imágenes de referencia como en HEVC y escalamiento de vectores de movimiento.

[0038] E. Alshina, A. Alshina, J.-H. Min, K. Choi, A. Saxena, M. Budagavi, "Known tools performance investigation for next generation video coding," ITU - Telecommunications Standardization Sector, STUDY GROUP 16 Question 6, Video Coding Experts Group (VCEG), VCEG-AZ05, junio. 2015, Warsaw, Poland (en adelante, "Alshina 1"), y A. Alshina, E. Alshina, T. Lee, "Bi-directional optical flow for improving motion compensation," Picture Coding Symposium (PCS), Nagoya, Japón, 2010 (en adelante, "Alshina 2") describen un método denominado flujo óptico bidireccional (BIO). El BIO se basa en el flujo óptico a nivel píxeles. De acuerdo con Alshina 1 y Alshina 2, BIO solo se aplica a bloques que tienen predicción directa e

indirecta. El BIO descrito en Alshina 1 y Alshina 2 se resume a continuación:

[0039] Dado un valor de píxel I_t en el tiempo t , la expansión de Taylor de primer orden del valor de píxel es

$$I_t = I_{t_0} + \frac{\partial I_{t_0}}{\partial t} (t - t_0) \quad (\text{A})$$

[0040] I_{t_0} está en la trayectoria de movimiento de I_t . Es decir, el movimiento de I_{t_0} a I_t se considera en la fórmula.

[0041] Bajo el supuesto de flujo óptico:

$$0 = \frac{dI}{dt} = \frac{\partial I}{\partial t} + \frac{\partial I}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \cdot \frac{\partial y}{\partial t}$$

$$\frac{\partial I}{\partial t} = -\frac{\partial I}{\partial x} \cdot \frac{\partial x}{\partial t} - \frac{\partial I}{\partial y} \cdot \frac{\partial y}{\partial t}$$

si $G_x = \frac{\partial I}{\partial x}$, $G_y = \frac{\partial I}{\partial y}$ (gradiente), y la ecuación (A) se convierte en

$$I_t = I_{t_0} - G_{x0} \cdot \frac{\partial x}{\partial t} \cdot (t - t_0) - G_{y0} \cdot \frac{\partial y}{\partial t} \cdot (t - t_0) \quad (\text{B})$$

[0042] Con respecto a $\frac{\partial x}{\partial t}$ y $\frac{\partial y}{\partial t}$ como la velocidad de movimiento, V_{x0} y V_{y0} se pueden usar para representarlas.

[0043] Entonces, la ecuación (B) se convierte en

$$I_t = I_{t_0} - G_{x0} \cdot V_{x0} \cdot (t - t_0) - G_{y0} \cdot V_{y0} \cdot (t - t_0) \quad (\text{C})$$

[0044] Al suponer una referencia directa en t_0 y una referencia indirecta en t_1 , y que $t_0 - t = t - t_1 = \Delta t = 1$

[0045] Entonces:

$$I_t = I_{t_0} - G_{x0} \cdot V_{x0} \cdot (t - t_0) - G_{y0} \cdot V_{y0} \cdot (t - t_0) = I_{t_0} + G_{x0} \cdot V_{x0} + G_{y0} \cdot V_{y0}$$

$$I_t = I_{t_1} - G_{x1} \cdot V_{x1} \cdot (t - t_1) - G_{y1} \cdot V_{y1} \cdot (t - t_1) = I_{t_1} - G_{x1} \cdot V_{x1} - G_{y1} \cdot V_{y1}$$

$$I_t = \frac{I_{t_0} + I_{t_1}}{2} + \frac{(G_{x0} \cdot V_{x0} - G_{x1} \cdot V_{x1}) + (G_{y0} \cdot V_{y0} - G_{y1} \cdot V_{y1})}{2} \quad (\text{D})$$

[0046] Además, se asume que $V_{x0} = V_{x1} = V_x$ y $V_{y0} = V_{y1} = V_y$ dado que el movimiento es a lo largo de la trayectoria. Entonces, la ecuación (D) se convierte en

$$I_t = \frac{I_{t_0} + I_{t_1}}{2} + \frac{(G_{x0} - G_{x1}) \cdot V_x + (G_{y0} - G_{y1}) \cdot V_y}{2} = \frac{I_{t_0} + I_{t_1}}{2} + \frac{\Delta G_x \cdot V_x + \Delta G_y \cdot V_y}{2} \quad (\text{E})$$

en donde $\Delta G_x = G_{x0} - G_{x1}$, $\Delta G_y = G_{y0} - G_{y1}$ se puede calcular en función de las referencias reconstruidas. Dado que $\frac{I_{t_0} + I_{t_1}}{2}$ es la bipredicción regular, $\frac{\Delta G_x \cdot V_x + \Delta G_y \cdot V_y}{2}$ se denomina en adelante desviación de BIO para mayor claridad.

[0047] V_x y V_y se derivan tanto en codificador como decodificador mediante la minimización de la siguiente distorsión:

$$\min \left\{ \sum_{bloque} \left((I_{t0} + G_{x0} \cdot V_x + G_{y0} \cdot V_y) - (I_{t1} - G_{x1} \cdot V_x - G_{y1} \cdot V_y) \right)^2 \right\}$$

$$= \min \left\{ \sum_{bloque} (\Delta I + (G_{x0} + G_{x1}) \cdot V_x + (G_{y0} + G_{y1}) \cdot V_y)^2 \right\}$$

[0048] Con V_x y V_y derivadas, la predicción final del bloque se calcula con la ecuación (E). V_x y V_y se denomina “movimiento de BIO” para mayor claridad.

[0049] En general, un codificador de video lleva a cabo BIO durante la compensación de movimiento. Es decir, después de que el codificador de video determina un vector de movimiento para un bloque actual, el codificador de video produce un bloque previsto para el bloque actual usando compensación de movimiento con respecto al vector de movimiento. En general, el vector de movimiento identifica la ubicación de un bloque de referencia con respecto al bloque actual en una imagen de referencia. Al llevar a cabo BIO, un codificador de video modifica el vector de movimiento por cada píxel para el bloque actual. Es decir, en vez de retirar cada píxel del bloque de referencia como una unidad de bloque, de acuerdo con BIO, el codificador de video determina modificaciones por píxel en el vector de movimiento para el bloque actual, y construye el bloque de referencia de manera que el bloque de referencia incluye píxeles de referencia identificados mediante el vector de movimiento y la modificación por píxel para el píxel correspondiente del bloque actual. De este modo, BIO se puede usar para producir un bloque de referencia más preciso para el bloque actual.

[0050] La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video 10 que puede utilizar técnicas para flujo óptico bidireccional. Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de video codificados que se desean decodificar posteriormente mediante un dispositivo de destino 14. En particular, el dispositivo fuente 12 proporciona los datos de video al dispositivo de destino 14 mediante un medio legible por computadora 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquier dispositivo del amplio rango de dispositivos, que incluyen computadoras de escritorio, notebooks (es decir, computadoras portátiles), tablets, decodificadores de televisión, microteléfonos, tales como los denominados “smartphones”, “smartpads”, televisiones, cámaras, dispositivos de visualización, transmisores de medios digitales, consolas de videojuegos, dispositivos de streaming de videos o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0051] El dispositivo de destino 14 puede recibir los datos de video codificados que se desean decodificar mediante el medio legible por computadora 16. El medio legible por computadora 16 puede comprender cualquier tipo de medio o dispositivo que puede mover los datos de video codificados del dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el medio legible por computadora 16 puede comprender un medio de comunicación que permite que el dispositivo fuente 12 transmita los datos de video codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de video codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal 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, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir routers, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación del dispositivo fuente 12 al dispositivo de destino 14.

[0052] En algunos ejemplos, los datos codificados pueden extraerse de la interfaz de salida 22 a un dispositivo de almacenamiento. De manera similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de los diversos medios de almacenamiento de datos distribuidos o que se pueden acceder localmente, tales 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. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermediario que pueda almacenar el video codificado generado por el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a los datos de video almacenados del dispositivo de almacenamiento mediante streaming o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 14. Los ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de video codificados a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal

inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, cable módem, etc.) o una combinación de ambos que sea adecuada para acceder a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados del dispositivo de almacenamiento puede ser una transmisión por streaming, una transmisión por descarga o una combinación de las dos.

[0053] Las técnicas de la presente descripción no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de video para diversas aplicaciones multimedia, tales como transmisiones por televisión inalámbrica, transmisiones por televisión por cable, transmisiones por televisión satelital, transmisiones de video por streaming en Internet, tales como streaming adaptativo dinámico en 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. En algunos ejemplos, el sistema 10 se puede configurar para permitir la transmisión de video unidireccional o bidireccional, a fin de permitir aplicaciones, tales como streaming de video, repetición de video, transmisión de video y/o videotelefonía.

[0054] En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye la fuente de video 18, el codificador de video 20 y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de video 30 y el dispositivo de visualización 32. De acuerdo con la presente divulgación, el codificador de video 20 del dispositivo fuente 12 se puede configurar para aplicar las técnicas para flujo óptico bidireccional. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de video de una fuente de video 18 externa, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo externo de visualización, en lugar de incluir un dispositivo integrado de visualización.

[0055] El sistema 10 ilustrado de la Figura 1 es solo un ejemplo. Las técnicas para flujo óptico bidireccional se pueden llevar a cabo mediante cualquier dispositivo de codificación y/o decodificación de video digital. Si bien las técnicas de la presente descripción se llevan a cabo generalmente mediante un dispositivo de codificación de video, también se pueden llevar a cabo mediante un codificador/decodificador de video, denominado generalmente "códec." Además, las técnicas de la presente descripción también se pueden llevar a cabo mediante un preprocesador de video. El

dispositivo fuente 12 y el dispositivo de destino 14 son solo ejemplos de los dispositivos de codificación, en donde el dispositivo fuente 12 genera datos de video codificados para la transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden operar de manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de video. Por lo tanto, el sistema 10 puede permitir la transmisión de video unidireccional o bidireccional entre los dispositivos de video 12, 14, por ejemplo, para streaming de video, repetición de video, transmisión de video o videotelefonía.

[0056] La fuente de video 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video capturado anteriormente y/o una interfaz de transmisión de video para recibir videos de un proveedor de contenidos de video. Como alternativa adicional, la fuente de video 18 puede generar datos basados en gráficos informáticos como el video fuente, o una combinación de video en vivo, video archivado y video generado por computadora. En algunos casos, si la fuente de video 18 es una cámara de video, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara fotográfica o videófonos. Sin embargo, como se mencionó anteriormente, las técnicas descritas en la presente descripción se pueden aplicar a la codificación de video en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas. En cada caso, el video capturado, precapturado o generado por computadora puede ser codificado por el codificador de video 20. La información de video codificada luego puede ser extraída por la interfaz de salida 22 en un medio legible por computadora 16.

[0057] El medio legible por computadora 16 puede incluir medios transitorios, tales como una transmisión de red inalámbrica o alámbrica, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como disco duro, unidad de disco flash, disco compacto, disco de video digital, disco Blu-ray, u otros medios legibles por computadora. En algunos ejemplos, un servidor de red (no se muestra) puede recibir datos de video codificados del dispositivo fuente 12 y proporcionar los datos de video codificados al dispositivo de destino 14, por ejemplo, mediante transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación para estampar discos, puede recibir datos de video codificados del dispositivo fuente 12 y producir un disco que contiene los datos de video codificados. Por lo tanto, el medio legible por computadora 16 puede incluir uno o más medios legibles por computadora de diversas formas, en

diversos ejemplos.

[0058] La interfaz de entrada 28 del dispositivo de destino 14 recibe información del medio legible por computadora 16. La información del medio legible por computadora 16 puede incluir información sintáctica definida por el codificador de video 20, que también es usada por el decodificador de video 30, que incluye elementos sintácticos que describen características y/o el procesamiento de datos de video. El dispositivo de visualización 32 muestra los datos de video decodificados a un usuario, y puede comprender cualquiera de los diversos dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico de emisión de luz (OLED), u otro tipo de dispositivo de visualización.

[0059] El codificador de video 20 y el decodificador de video 30 pueden funcionar de acuerdo con uno o más estándares de codificación de video, tal como ITU-T H.264/AVC (codificación de video avanzada) o codificación de video de alta eficacia (HEVC), también denominada ITU-T H.265. H.264 se describe en International Telecommunication Union, “Advanced video coding for generic audiovisual services,” SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services – Coding of moving video, H.264, junio de 2011. H.265 se describe en International Telecommunication Union, “High efficiency video coding,” SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services – Coding of moving video, abril de 2015. Las técnicas de la presente divulgación también se pueden aplicar a cualquier otro estándar de codificación de video previo o futuro como una herramienta de codificación eficaz.

[0060] Otros estándares de codificación de video incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual y las extensiones de codificación de video escalable (SVC) y codificación de video multivista (MVC) de H.264, así como también las extensiones de HEVC, tal como extensión de rango, extensión multivista (MV-HEVC) y extensión escalable (SHVC). En abril de 2015, el grupo de expertos en codificación de video (VCEG) comenzó un nuevo proyecto de investigación que tiene como objetivo una próxima generación de estándares de codificación de video. El software de referencia se denomina HM-KTA.

[0061] En la actualidad, ITU-T VCEG (Q6/16) y ISO/IEC MPEG (JTC 1/SC 29/WG 11) estudian la necesidad potencial de estandarización de tecnología de codificación de video futura con una capacidad de compresión que excede significativamente la del

estándar de HEVC actual (que incluye extensiones actuales de HEVC y extensiones a corto plazo para codificación de contenido de pantalla y codificación de alto rango dinámico). Los grupos trabajan juntos en esta actividad de exploración en un esfuerzo de colaboración conjunta conocido como equipo de exploración de video conjunta (JVET) para evaluar diseños de tecnología de compresión propuestos por los expertos en la materia. El JVET se reunió por primera vez del 19 al 21 de octubre de 2015. La última versión del *software* de referencia, es decir, modelo de exploración conjunta 3 (JEM 3) se puede descargar de:

https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-4.0/

La descripción de un algoritmo del modelo de prueba de exploración conjunta 3 (JEM3) se puede denominar JVET-D1001.

[0062] Determinadas técnicas de codificación de video, tales como aquellas de H.264 y HEVC que se relacionan con las técnicas de la presente divulgación, se describen en la presente divulgación. Algunas técnicas de la presente divulgación se pueden describir en referencia a H.264 y/o HEVC para ayudar en el entendimiento, pero las técnicas descritas no se limitan necesariamente a H.264 o HEVC y se pueden usar de manera conjunta con otros estándares de codificación y otras herramientas de codificación.

[0063] Si bien no se muestra en la Figura 1, en algunos aspectos, el codificador de video 20 y el decodificador de video 30 pueden estar integrados, cada uno, con un codificador y un decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro *hardware* y *software*, para la codificación de audio y video en una corriente de datos común o corrientes de datos separadas. Si correspondiera, las unidades MUX-DEMUX pueden adecuarse al protocolo de multiplexor ITU H.223, o a otros protocolos, tales como el protocolo de datagrama de usuario (UDP).

[0064] En HEVC y otras especificaciones de codificación de video, una secuencia de video generalmente incluye una serie de imágenes. Las imágenes también se pueden denominar “cuadros”. Una imagen puede incluir tres matices de muestra, indicadas como S_L , S_{Cb} , y S_{Cr} . S_L es una matriz de dos dimensiones (por ejemplo, un bloque) de muestras de luminancia. S_{Cb} es una matriz de dos dimensiones de muestras de crominancia Cb. S_{Cr} es una matriz de dos dimensiones de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en la presente muestras de croma. En otros ejemplos, una imagen puede ser monocromática y solo puede incluir una matriz de muestras de luminancia.

[0065] A fin de generar una representación codificada de una imagen, el codificador

de video 20 puede generar un conjunto de unidades de decodificación en árbol (CTU). Cada una de las CTU puede comprender un bloque de codificación en árbol de muestras de luminancia, dos bloques de codificación en árbol correspondientes de muestras de crominancia y estructuras sintácticas que se usan para codificar las muestras de los bloques de codificación en árbol. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un solo bloque de codificación en árbol y estructuras sintácticas que se usan para codificar las muestras del bloque de codificación en árbol. Un bloque de codificación en árbol puede ser un bloque de muestras NxN. Una CTU también puede denominarse “bloque de árbol” o “unidad de codificación más grande” (LCU). Las CTU de HEVC pueden ser ampliamente análogas a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no se limita necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un segmento puede incluir un número entero de CTU ordenado de manera consecutiva en un orden de exploración de trama.

[0066] Un CTB contiene un árbol cuaternario cuyos nodos son unidades de codificación. El tamaño de una CTB puede variar de 16x16 a 64x64 en el perfil *main* de HEVC (aunque, técnicamente, se pueden permitir tamaños de CTB 8x8). Sin embargo, una unidad de codificación (CU) podría tener el mismo tamaño que un CTB, y podría ser tan pequeña como 8x8. Cada unidad de codificación se codifica con un modo. Cuando una CU está intercodificada, la CU puede particionar aún más en 2 o 4 unidades de predicción (PU) o volverse una sola PU cuando no se puede aplicar mayor partición. Cuando dos PU están presentes en una CU, las dos PU pueden ser rectángulos de igual tamaño o dos rectángulos, uno de los cuales tiene el tamaño de $\frac{1}{4}$ y el otro de $\frac{3}{4}$ de la CU.

[0067] A fin de generar una CTU codificada, el codificador de video 20 puede realizar recursivamente la partición del árbol cuaternario en los bloques de codificación en árbol de una CTU para dividir los bloques de codificación en árbol en bloques de codificación, de aquí el nombre “unidades de decodificación en árbol”. Un bloque de codificación puede ser un bloque de muestras NxN. Una CU puede comprender un bloque de codificación de muestras de luminancia y dos bloques de codificación correspondientes de muestras de crominancia de una imagen que tiene una matriz de muestras de luminancia, una matriz de muestras de Cb y una matriz de muestras de Cr, y estructuras sintácticas que se usan para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color

separados, una CU puede comprender un solo bloque de codificación y las estructuras sintácticas que se usan para codificar las muestras del bloque de codificación.

[0068] El codificador de video 20 puede particionar un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras sobre las cuales se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luminancia, dos bloques de predicción correspondientes de muestras de crominancia y estructuras sintácticas que se usan para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un solo bloque de predicción y estructuras sintácticas que se usan para predecir el bloque de predicción. El codificador de video 20 puede generar bloques predictivos de luminancia, Cb y Cr para bloques de predicción de luminancia, Cb y Cr de cada PU de la CU.

[0069] El codificador de video 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de video 20 utiliza la intrapredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU en función de muestras decodificadas de la imagen asociada a la PU. Si el codificador de video 20 utiliza la interpredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU en función de muestras decodificadas de una o más imágenes distintas de la imagen asociada a la PU. Cuando la CU está intercodificada, puede haber un conjunto de información de movimiento presente para cada PU. Asimismo, cada PU se puede codificar con un modo de interpredicción único para derivar el conjunto de información de movimiento.

[0070] Después de que el codificador de video 20 genera los bloques predictivos de luminancia, Cb y Cr para una o más PU de una CU, el codificador de video 20 puede generar un bloque residual de luminancia para la CU. Cada muestra en el bloque residual de luminancia de la CU indica una diferencia entre una muestra de luminancia en uno de los bloques de luminancia predictivos de la CU y una muestra correspondiente en el bloque de codificación de luminancia original de la CU. Asimismo, el codificador de video 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques de Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El

codificador de video 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques de Cr predictivos de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

[0071] Además, el codificador de video 20 puede usar la partición del árbol cuaternario para descomponer los bloques residuales de luminancia, Cb y Cr de una CU en uno o más bloques de la transformada de luminancia, Cb y Cr. Un bloque de la transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras sobre las cuales se aplica la misma transformada. Una unidad de la transformada (TU) de una CU puede comprender un bloque de la transformada de muestras de luminancia, dos bloques de la transformada correspondientes de muestras de crominancia y estructuras sintácticas que se usan para transformar las muestras del bloque de la transformada. Por lo tanto, cada TU de una CU se puede asociar a un bloque de la transformada de luminancia, un bloque de la transformada de Cb y un bloque de la transformada de Cr. El bloque de la transformada de luminancia asociado a la TU puede ser un subbloque del bloque residual de luminancia de la CU. El bloque de la transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de la transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un solo bloque de la transformada y las estructuras sintácticas que se usan para transformar las muestras del bloque de la transformada.

[0072] El codificador de video 20 puede aplicar una o más transformadas a un bloque de la transformada de luminancia de una TU para generar un bloque de coeficiente de luminancia para la TU. Un bloque de coeficiente puede ser una matriz de coeficientes de la transformada bidimensional. Un coeficiente de la transformada puede ser una cantidad escalar. El codificador de video 20 puede aplicar una o más transformadas a un bloque de la transformada de Cb de una TU para generar un bloque de coeficiente de Cb para la TU. El codificador de video 20 puede aplicar una o más transformadas a un bloque de la transformada de Cr de una TU para generar un bloque de coeficiente de Cr para la TU.

[0073] Después de generar un bloque de coeficiente (por ejemplo, un bloque de coeficiente de luminancia, un bloque de coeficiente de Cb o un bloque de coeficiente de Cr), el codificador de video 20 puede cuantificar el bloque de coeficiente. En general, la cuantificación se refiere a un proceso en donde los coeficientes de la

transformada se cuantifican para reducir posiblemente la cantidad de datos que se usan para representar los coeficientes de la transformada, lo que proporciona una mayor compresión. Después de que el codificador de video 20 cuantifica un bloque de coeficiente, el codificador de video 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de la transformada cuantificados. Por ejemplo, el codificador de video 20 puede realizar la codificación aritmética binaria adaptativa según el contexto (CABAC) en los elementos sintácticos que indican los coeficientes de la transformada cuantificados.

[0074] El codificador de video 20 puede extraer una corriente de bits que incluye una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. La corriente de bits puede comprender una secuencia de unidades NAL. Una unidad de NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad de NAL y bytes que contienen esos datos en forma de una RBSP intercalada según sea necesario con bits de prevención de emulación. Cada una de las unidades de NAL incluye un encabezado de la unidad NAL y encapsula una RBSP. El encabezado de la unidad de NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad de NAL. El código de tipo de unidad de NAL especificado por el encabezado de la unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una unidad sintáctica que contiene un número entero de bytes que está encapsulado dentro de una unidad de NAL. En algunos ejemplos, una RBSP no incluye ningún bit.

[0075] Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un PPS, un segundo tipo de unidad de NAL puede encapsular una RBSP para un segmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para mensajes SEI, etcétera. Las unidades de NAL que encapsulan RBSP para la codificación de datos de video (a diferencia de las RBSP para los conjuntos de parámetros y mensajes SEI) se pueden denominar unidades de NAL de VCL.

[0076] El decodificador de video 30 puede recibir una corriente de bits generada por el codificador de video 20. Además, el decodificador de video 30 puede analizar la corriente de bits para obtener elementos sintácticos de la corriente de bits. El decodificador de video 30 puede reconstruir las imágenes de los datos de video en función de, al menos en parte, los elementos sintácticos obtenidos de la corriente de bits. Generalmente, el proceso para reconstruir los datos de video es recíproco al proceso realizado por el codificador de video 20. Además, el decodificador de video 30

puede cuantificar de manera inversa los bloques de coeficiente asociados a las TU de una CU actual. El decodificador de video 30 puede realizar transformadas inversas en los bloques de coeficiente para reconstruir bloques de la transformada asociados a las TU de la CU actual. El decodificador de video 30 puede reconstruir los bloques de codificación de la CU actual agregando las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de la transformada de las TU de la CU actual. Al reconstruir los bloques de codificación para cada CU de una imagen, el decodificador de video 30 puede reconstruir la imagen.

[0077] De acuerdo con las técnicas de la presente divulgación, el codificador de video 20 y/o el codificador de video 30 adicionalmente pueden llevar a cabo técnicas de BIO durante la compensación de movimiento como se indica en mayor detalle a continuación.

[0078] El codificador de video 20 y el decodificador de video 30 se pueden implementar, cada uno, como cualquiera de los diversos circuitos codificadores o decodificadores adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados para aplicaciones específicas (ASIC), matrices de compuerta programable en campo (FPGA), circuito lógico discreto, software, hardware, firmware o cualquier combinación de estos. El codificador de video 20 y el decodificador de video 30 pueden estar incluidos, cada uno, en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador de video (CÓDEC) combinado. Un dispositivo que incluye codificador de video 20 y/o decodificador de video 30 puede comprender un circuito integrado, un microprocesador, y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

[0079] La Figura 2 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) unilateral como un algoritmo de comparación de bloques (BMA) realizado para la conversión ascendente de la velocidad de cuadro de movimiento compensado (MC-FRUC). En general, un codificador de video (tal como codificador de video 20 o decodificador de video 30) realiza ME unilateral para obtener vectores de movimiento (MV), tales como MV 112, al buscar el bloque de mejor coincidencia (por ejemplo, bloque de referencia 108) del cuadro de referencia 102 para el bloque actual 106 del cuadro actual 100. Luego, el codificador de video interpola un bloque interpolado 110 junto a la trayectoria de movimiento del vector de movimiento 112 en el cuadro interpolado 104. Es decir, en el ejemplo de la Figura 2, el vector de movimiento 112 pasa a través de los puntos medios del bloque actual 106, el bloque

de referencia 108 y el bloque interpolado 110.

[0080] Como se muestra en la Figura 2, tres bloques en tres cuadros participan del seguimiento de la trayectoria de movimiento. Aunque el bloque actual 106 en el cuadro actual 100 pertenece a un bloque codificado, el bloque de mejor coincidencia en el cuadro de referencia 102 (es decir, bloque de referencia 108) puede no pertenecer completamente a un bloque codificado (es decir, el bloque de mejor coincidencia puede no entrar en el límite del bloque codificado, pero sí superponerse con ese límite). Asimismo, el bloque interpolado 110 en el cuadro interpolado 104 puede no pertenecer completamente a un bloque codificado. En consecuencia, pueden surgir regiones superpuestas de los bloques y regiones vacías (agujeros) en el cuadro interpolado 104.

[0081] Para manejar las superposiciones, los algoritmos FRUC simples pueden implicar promediar y sobrescribir los píxeles superpuestos. Asimismo, los agujeros pueden cubrirse con los valores de píxeles de un cuadro actual o de referencia. Sin embargo, estos algoritmos pueden producir artefactos de compresión e imagen borrosa. Por lo tanto, la segmentación del campo de movimiento, la extrapolación sucesiva usando la transformada discreta de Hartley y la restauración de imagen se pueden usar para tratar agujeros y superposiciones sin aumentar los artefactos de compresión y la imagen borrosa.

[0082] La Figura 3 es un diagrama conceptual que ilustra un ejemplo de ME bilateral como un BMA realizado para la MC-FRUC. ME bilateral es otra solución (en MC-FRUC) que se puede usar para evitar los problemas causados por las superposiciones y los agujeros. Un codificador de video (tal como codificador de video 20 y/o decodificador de video 30) que realiza ME bilateral obtiene MV 132, 134 que pasan a través de un bloque interpolado 130 del cuadro interpolado 124 (que es intermedio entre el cuadro actual 120 y el cuadro de referencia 122) usando simetría temporal entre el bloque actual 126 del cuadro actual 120 y el bloque de referencia 128 del cuadro de referencia 122. Como resultado, el codificador de video no produce superposiciones y agujeros en el cuadro interpolado 124. Suponiendo que el bloque actual 126 es un bloque que el codificador de video codifica en un determinado orden, por ejemplo, como en el caso de codificación de video, una secuencia de tales bloques cubriría la imagen intermedia en su totalidad sin superposición. Por ejemplo, en el caso de codificación de video, los bloques se pueden procesar en el orden de decodificación. Por lo tanto, un método de este tipo sería más adecuado si las ideas de FRUC se pudieran considerar en una estructura de marco de codificación de video.

[0083] S.-F. Tu, O. C. Au, Y. Wu, E. Luo y C.-H. Yeun, “A Novel Framework for Frame Rate Up Conversion by Predictive Variable Block-Size Motion Estimated Optical Flow,” International Congress on Image Signal Processing (CISP), 2009 describieron una estimación de movimiento a nivel de bloque híbrido y un método de flujo óptico a nivel del píxel para la conversión ascendente de la velocidad de cuadro. Tu estableció que el panorama híbrido era mejor que cualquiera de los métodos individuales.

[0084] En el estándar HEVC, hay dos modos de interpredicción, denominados modo de fusión (en donde el modo de salto se considera un caso especial de modo de fusión) y modo de predicción avanzada de vectores de movimiento (AMVP). Ya sea en modo de AMVP o de fusión, un codificador de video mantiene una lista de MV candidatos para múltiples predictores de vectores de movimiento. Un codificador de video determina que los vectores de movimiento para una PU en particular, así como los índices de referencia en el modo de fusión, seleccionen un candidato de la lista de MV candidatos.

[0085] En HEVC, la lista de MV candidatos contiene hasta 5 candidatos para el modo de fusión y solo 2 candidatos para el modo de AMVP. Otros estándares de codificación pueden incluir más o menos candidatos. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento que corresponden tanto a las listas de imágenes de referencia (lista 0 y lista 1) como a los índices de referencia. Un decodificador de video recibe un candidato de fusión identificado por un índice de fusión, y el decodificador de video predice una PU actual usando las imágenes de referencia identificadas y los vectores de movimiento. Sin embargo, para el modo de AMVP, para cada posible dirección de predicción de la lista 0 o la lista 1, debe señalizarse explícitamente un índice de referencia, junto con un índice de predictores de MV (MVP) para la lista de MV candidatos, debido a que el AMVP candidato contiene únicamente un vector de movimiento. En el modo de AMVP, los vectores de movimiento previstos se pueden refinar adicionalmente.

[0086] Un candidato de fusión corresponde a un conjunto completo de información de movimiento mientras que un candidato de AMVP contiene solamente un vector de movimiento para una dirección de predicción y un índice de referencia específicos. Los candidatos de ambos modos derivan, de manera similar, de los mismos bloques vecinos espaciales y temporales.

[0087] La Figura 4A muestra MV candidatos vecinos espaciales para el modo de fusión, y la Figura 4B muestra MV candidatos vecinos espaciales para los modos de AMVP. Los MV candidatos espaciales derivan de los bloques vecinos que se muestran

en las Figuras 4A y 4B, para una PU específica (PU_0), aunque los métodos que generan los candidatos de los bloques difieren de los modos de fusión y de AMVP.

[0088] En el modo de fusión, se pueden derivar hasta 4 MV candidatos espaciales con el orden que se muestra en la Figura 4A. El orden es el siguiente: izquierdo (0), superior (1), superior derecho (2), inferior izquierdo (3) y superior izquierdo (4), como se muestra en la Figura 4A. Si todos los MV candidatos espaciales 0-3 están disponibles y son únicos, entonces el codificador de video puede no incluir información de movimiento para el bloque superior izquierdo en la lista de candidatos. No obstante, si uno o más de los MV candidatos espaciales 0-3 no están disponibles y no son únicos, entonces el codificador de video puede incluir información de movimiento para el bloque superior izquierdo en la lista de candidatos.

[0089] En el modo de AVMP, los bloques vecinos se dividen en dos grupos: grupo izquierdo que consiste en el bloque 0 y 1, y el grupo superior que consiste en los bloques 2, 3 y 4 como se muestra en la Figura 4B. Para cada grupo, el vector de candidato posible en un bloque vecino que hace referencia a la misma imagen de referencia a la indicada por el índice de referencia señalado tiene la máxima prioridad de ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte hacia la misma imagen de referencia. Por lo tanto, si no se puede encontrar ese candidato, el primer candidato disponible puede escalarse para formar el candidato final, por ende, las diferencias de distancia temporal se pueden compensar.

[0090] La Figura 5A muestra un ejemplo de un TMVP candidato, y la Figura 5B muestra un ejemplo de escalamiento de MV. Un predictor temporal de vectores de movimiento (TMVP) candidato, si se genera y está disponible, se agrega a la lista de MV candidatos después de vectores de movimiento candidatos espaciales. El proceso de derivación de vectores de movimiento para un TMVP candidato es el mismo para los modos de fusión y de AMVP, sin embargo, el índice de referencia objetivo para el TMVP candidato en el modo de fusión siempre está configurado en 0.

[0091] La ubicación primaria del bloque para la derivación de un TMVP candidato es el bloque inferior derecho fuera de la PU colocalizada como se muestra en la Figura 5A como bloque "T", para compensar la desviación en los bloques superiores e izquierdos usados para generar candidatos vecinos espaciales. Sin embargo, si ese bloque está localizado fuera de la fila de CTB actual, o la información de movimiento no está disponible, el bloque se sustituye con un bloque central de la PU.

[0092] El vector de movimiento para TMVP candidato se deriva de la PU colocalizada

de la imagen colocada, que se indica en el nivel del segmento. El vector de movimiento para la PU colocada se denomina MV colocado. De manera similar al modo directo temporal en AVC, para derivar el vector de movimiento del TMVP candidato, el MV colocado necesita escalarse para compensar las diferencias de distancia temporal, como se muestra en las Figuras 5B.

[0093] HEVC también usa escalamiento de vectores de movimiento. Se asume que el valor de los vectores de movimiento es proporcional a la distancia de las imágenes en el tiempo de presentación. Un vector de movimiento se asocia a dos imágenes, la imagen de referencia y la imagen que contiene el vector de movimiento (a saber, la imagen contenedora). Cuando un vector de movimiento se usa para predecir otro vector de movimiento, la distancia de la imagen contenedora y la imagen de referencia se calcula en función de los valores del POC.

[0094] Para un vector de movimiento que se desea predecir, la imagen contenedora asociada para el vector de movimiento y la imagen de referencia pueden ser diferentes. Por lo tanto, se calcula una nueva distancia (en función del POC), y el vector de movimiento se escala con base en estas dos distancias POC. Para un candidato vecino espacial, las imágenes contenedoras para los dos vectores de movimiento son iguales, mientras que las imágenes de referencia son diferentes. En HEVC, el escalamiento de vectores de movimiento se aplica a ambos TMVP y AMVP para vecinos espaciales y temporales candidatos.

[0095] HEVC también usa generación de vectores de movimiento candidatos artificiales. Si una lista de vectores de movimiento candidatos no está completa, se generan vectores de movimiento candidatos artificiales y se insertan al final de la lista hasta que todas las entradas disponibles en la lista de vectores de movimiento candidatos tengan un candidato. En el modo de fusión, existen dos tipos de MV candidatos artificiales: candidato combinado derivado solo para los segmentos B y candidatos cero que se usan solo para AMVP si el primer tipo no proporciona suficientes candidatos artificiales. Para cada par de candidatos que ya se encuentran en la lista de candidatos y tienen información de movimiento necesaria, los vectores de movimiento candidatos combinados bidireccionales se derivan mediante una combinación del vector de movimiento del primer candidato, haciendo referencia a una imagen en la lista 0, y el vector de movimiento de un segundo candidato, haciendo referencia a una imagen en la lista 1.

[0096] HEVC también utiliza un proceso de reducción para la inserción de candidatos. Los candidatos de diferentes bloques puede ser iguales, lo que disminuye la eficacia

de una lista de candidatos de fusión/AMVP. Un proceso de reducción puede aplicarse para resolver este problema. Un proceso de reducción compara un candidato con otros en la lista de candidatos actual para evitar insertar un candidato idéntico. Para reducir la complejidad, solo una cantidad limitada de procesos de reducción pueden aplicarse en lugar de comparar cada uno posible con todos los existentes. Como un ejemplo, un codificador de video puede aplicar un proceso de reducción a los candidatos vecinos espaciales y temporales pero no a candidatos generados artificialmente.

[0097] Se describirán los aspectos del flujo óptico bidireccional en JEM. La Figura 6 muestra un ejemplo de trayectoria de flujo óptico. BIO utiliza refinamiento de movimiento en píxeles que se lleva a cabo por encima de la compensación de movimiento en bloques en caso de bipredicción. Puesto que BIO compensa el movimiento fino dentro del bloque, la habilitación de BIO puede efectivamente dar como resultado un aumento del tamaño de bloque para la compensación de movimiento. El refinamiento de movimiento al nivel de la muestra no requiere una búsqueda exhaustiva o señalización, sino que, en su lugar, utiliza una ecuación explícita que proporciona vectores de movimiento finos para cada muestra.

[0098] Si $I^{(k)}$ es el valor de luminancia de k ($k=0, 1$) de referencia después del movimiento del bloque de compensación, $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ son componentes horizontales y verticales del gradiente $I^{(k)}$, respectivamente. Suponiendo que el flujo óptico es válido, el campo del vector de movimiento (v_x, v_y) se obtiene mediante una ecuación

$$\partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad (1)$$

[0099] Si se combina la ecuación de flujo óptico con interpolación de Hermite para la trayectoria de movimiento de cada muestra se obtiene un polinomio único de tercer orden que coincide con ambos valores funcionales $I^{(k)}$ y derivadas $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ en los extremos. El valor de este polinomio en $t=0$ es la predicción de BIO:

$$pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad (2)$$

[0100] En la presente, τ_0 y τ_1 indican la distancia a los cuadros de referencia como se muestra en la Figura 6. Las distancias τ_0 y τ_1 se calculan en función del POC para Ref0 y Ref1: $\tau_0 = \text{POC}(\text{actual}) - \text{POC}(\text{Ref0})$, $\tau_1 = \text{POC}(\text{Ref1}) - \text{POC}(\text{actual})$. Si

ambas predicciones provienen de la misma dirección (ambas del pasado o ambas del futuro) los signos son diferentes $\tau_0 \cdot \tau_1 < 0$. En este caso, BIO se aplica solo si la predicción no proviene del mismo momento temporal ($\tau_0 \neq \tau_1$), ambas regiones de referencia tienen movimiento distinto a cero ($MVx_0, MVy_0, MVx_1, MVy_1 \neq 0$) y los vectores de movimiento del bloque son proporcionales a la distancia temporal ($MVx_0/MVx_1 = MVy_0/MVy_1 = -\tau_0/\tau_1$).

[0101] El campo del vector de movimiento (v_x, v_y) , también denominado cantidad de movimiento de BIO, se determina al minimizar la diferencia Δ entre los valores en los puntos A y B (planos de intersección de trayectoria de movimiento y cuadro de referencia en la Figura 6). El modelo solo usa el primer término lineal de expansión de Taylor local para Δ :

$$\Delta = (I^{(0)} - I^{(1)}_0 + v_x (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) + v_y (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)) \quad (3)$$

[0102] Todos los valores en (1) dependen de la ubicación de la muestra (i', j') , que hasta el momento se omitió. Suponiendo que el movimiento es consistente en el entorno local, entonces Δ dentro de $(2M+1) \times (2M+1)$ la ventana cuadrada Ω centrada en un punto actualmente previsto (i, j) puede minimizarse de la siguiente manera:

$$(v_x, v_y) = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{[i', j'] \in \Omega} \Delta^2[i', j'] \quad (4)$$

[0103] Para este problema de optimización, se puede usar una solución simplificada de realizar primero una minimización en dirección vertical y, luego, horizontal, lo que da como resultado:

$$v_x = (s_1 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_3}{(s_1 + r)} \right) : 0 \quad (5)$$

$$v_y = (s_5 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5 + r)} \right) : 0 \quad (6)$$

en donde,

$$\begin{aligned} s_1 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x); \\ s_2 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\ s_5 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (7)$$

[0104] A fin de evitar la división por cero o valores muy pequeños, se introducen los parámetros de regularización r y m en las ecuaciones (2), (3).

$$r = 500 \cdot 4^{d-8} \quad (8)$$

$$m = 700 \cdot 4^{d-8} \quad (9)$$

En la presente, d es la profundidad de bits del video de entrada.

[0105] En algunos casos, el refinamiento de MV de BIO puede no resultar confiable debido al ruido o al movimiento irregular. Por lo tanto, en BIO, la magnitud de refinamiento de MV se asocia a cierto umbral th_{BIO} . El valor umbral se determina en función de si todas las imágenes de referencia de la imagen actual provienen todas de una dirección. Si todas las imágenes de referencia de las imágenes actuales de la imagen actual provienen de una dirección, el valor del umbral puede establecerse en $12 \times 2^{14-d}$; de otro modo, el umbral puede establecerse en $12 \times 2^{13-d}$.

[0106] Los gradientes para BIO se calculan al mismo tiempo con la interpolación de compensación de movimiento usando operaciones consistentes con el proceso de compensación de movimiento de HEVC (FIR 2D separable). La entrada para esta FIR 2D separable es la misma muestra del cuadro de referencia que para el proceso de compensación de movimiento y posición fraccional ($fracX$, $fracY$) de acuerdo con la parte fraccional del vector de movimiento del bloque. En el caso del gradiente horizontal $\partial I / \partial x$, la señal se interpoló primero verticalmente usando *BIOfilterS* correspondiente a la posición fraccional $fracY$ con desplazamiento de escalamiento inverso $d-8$, luego se aplicó el filtro de gradiente *BIOfilterG* en dirección horizontal correspondiente a la posición fraccional $fracX$ con desplazamiento de escalamiento inverso por $18-d$. En caso del gradiente vertical $\partial I / \partial y$, el primer filtro de gradiente se aplica de manera vertical usando *BIOfilterG* correspondiente a la posición fraccional $fracY$ con desplazamiento de escalamiento inverso $d-8$, luego se realiza desplazamiento de señal usando *BIOfilterS* en dirección horizontal correspondiente a la posición fraccional $fracX$ con desplazamiento de escalamiento inverso $18-d$. La longitud del filtro de interpolación para el cálculo de los gradientes *BIOfilterG* y el desplazamiento de señal *BIOfilterF* es menor (6-tap) a fin de mantener una complejidad razonable. La Tabla 1 muestra los filtros usados para el cálculo de los gradientes para diferentes posiciones fraccionales del vector de movimiento del bloque en BIO. La Tabla 2 muestra los filtros de interpolación usados para la generación de señales de predicción en BIO.

[0107] La Figura 7 muestra un ejemplo del cálculo de gradiente para un bloque 8x4. Para los bloques 8x4, un codificador de video busca los predictores de movimiento compensado y calcula los gradientes HOR/VER de todos los píxeles en el bloque actual así como las dos líneas externas de píxeles, puesto que para resolver v_x y v_y

para cada píxel se necesitan los valores de gradientes HOR/VER y los predictores de movimiento compensado de los píxeles en la ventana Ω centrada en cada píxel como se muestra en la ecuación (4). En JEM, el tamaño de la ventana se establece en 5x5. Por lo tanto, un codificador de video necesita buscar los predictores de movimiento compensado y calcular los gradientes de las dos líneas externas de píxeles.

Tabla 1: Filtros para el cálculo de gradientes en BIO

Posición de pel fraccionario	Filtro de interpolación para gradiente (BIOfilterG)
0	{ 8, -39, -3, 46, -17, 5}
1/16	{ 8, -32, -13, 50, -18, 5}
1/8	{ 7, -27, -20, 54, -19, 5}
3/16	{ 6, -21, -29, 57, -18, 5}
1/4	{ 4, -17, -36, 60, -15, 4}
5/16	{ 3, -9, -44, 61, -15, 4}
3/8	{ 1, -4, -48, 61, -13, 3}
7/16	{ 0, 1, -54, 60, -9, 2}
1/2	{ 1, 4, -57, 57, -4, 1}

Tabla 2: Filtros de interpolación para generación de señal de predicción en BIO

Posición de pel fraccionario	Filtro de interpolación para señal de predicción (BIOfilterS)
0	{ 0, 0, 64, 0, 0, 0 }
1/16	{ 1, -3, 64, 4, -2, 0 }
1/8	{ 1, -6, 62, 9, -3, 1 }
3/16	{ 2, -8, 60, 14, -5, 1 }
1/4	{ 2, -9, 57, 19, -7, 2 }
5/16	{ 3, -10, 53, 24, -8, 2 }
3/8	{ 3, -11, 50, 29, -9, 2 }
7/16	{ 3, -11, 44, 35, -10, 3 }
1/2	{ 1, -7, 38, 38, -7, 1 }

[0108] En JEM, BIO se aplica a todos los bloques previstos bidireccionales cuando las dos predicciones provienen de imágenes de referencia distintas. Cuando LIC se habilita para CU, BIO se inhabilita.

[0109] La Figura 8 muestra un ejemplo de BIO modificado para el bloque 8x4 propuesto en JVET-D0042. En el 4.º Encuentro del JVET, una propuesta JVET-D0042 (A. Alshina, E. Alshina, “AHG6: On BIO memory bandwidth”, JVET-D0042, octubre 2016) se presentó para modificar las operaciones BIO y reducir el ancho de banda de acceso de la memoria. En esta propuesta, no se necesita ningún predictor de movimiento compensado ni valor de gradiente para los píxeles fuera del bloque actual. Además, la resolución de vx y vy para cada píxel se modifica para usar los predictores de movimiento compensado y los valores de gradiente de todos los píxeles en el bloque actual como se muestra en la Figura 8. En otras palabras, la ventana cuadrada Ω en la ecuación (4) se modifica a una ventana que es igual al bloque actual. Además, un factor de ponderación $w(i',j')$ se considera para derivar vx y vy. El $w(i',j')$ es una función de la posición del píxel central (i,j) y la posición de los píxeles (i',j') dentro de la ventana.

$$\begin{aligned}
 s_1 &= \sum_{[i',j'] \in \Omega} w(i',j') (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 = \sum_{[i',j'] \in \Omega} w(i',j') (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x); \\
 s_2 &= \sum_{[i',j'] \in \Omega} w(i',j') (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\
 s_5 &= \sum_{[i',j'] \in \Omega} w(i',j') (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 = \sum_{[i',j'] \in \Omega} w(i',j') (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)
 \end{aligned} \tag{10}$$

[0110] Se describirán aspectos de la compensación de movimiento de bloques superpuestos (OBMC) en JEM. OBMC se usó en las primeras generaciones de estándares de video, por ejemplo, como en H.263. En JEM, OBMC se realiza para todos los límites de los bloques de movimiento compensado (MC) excepto por los límites derecho e inferior de una CU. Además, OBMC puede aplicarse para los

componentes de luminancia y crominancia. En JEM, un bloque de MC corresponde a un bloque de codificación. Cuando una CU se codifica con un modo sub-CU (incluye fusión de sub-CU, modo afín y modo FRUC, como se describe en J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, “Algorithm Description of Joint Exploration Test Model 4,” JVET-D1001, octubre 2016), cada subbloque de la CU es un bloque de MC. Para procesar límites de CU de modo uniforme, OBMC se lleva a cabo a nivel subbloque para todos los límites del bloque de MC, en donde el tamaño del subbloque se establece en 4x4, como se ilustra en las Figuras 9A y 9B.

[0111] Cuando OBMC aplica al subbloque actual, además de vectores de movimiento actuales, también se usan vectores de movimiento de 4 subbloques vecinos conectados, en caso de estar disponibles y no ser idénticos al vector de movimiento actual, para derivar el bloque de predicción para el subbloque actual. Estos bloques de predicción múltiples con base en múltiples vectores de movimiento se combinan para generar la señal de predicción final del subbloque actual.

[0112] Como se muestra en la Figura 10, el bloque de predicción con base en vectores de movimiento de un subbloque vecino se denomina P_N , en donde N indica un índice para los subbloques vecinos *superior*, *inferior*, *izquierdo* y *derecho* y el bloque de predicción con base en vectores de movimiento del subbloque actual se denomina P_C . Cuando P_N se basa en la información de movimiento de un subbloque vecino que contiene la misma información de movimiento que el subbloque actual, OBMC no se realiza para P_N . De otro modo, cada píxel de P_N se agrega al mismo píxel en P_C , es decir, 4 filas/columnas de P_N se agregan a P_C . Los factores de ponderación $\{1/4, 1/8, 1/16, 1/32\}$ se usan para P_N y los factores de ponderación $\{3/4, 7/8, 15/16, 31/32\}$ se usan para P_C . Los bloques de MC pequeños son la excepción, (es decir, cuando la altura o el ancho del bloque de codificación equivale a 4 o una CU se codifica con modo sub-CU), para los que solo dos filas/columnas de P_N se agregan a P_C . En este caso, los factores de ponderación $\{1/4, 1/8\}$ se usan para P_N y los factores de ponderación $\{3/4, 7/8\}$ se usan para P_C . Para P_N generados con base en vectores de movimiento de subbloque verticalmente (horizontalmente) vecino, los píxeles en la misma fila (columna) de P_N se agregan a P_C con un mismo factor de ponderación. BIO también puede aplicarse para la derivación del bloque de predicción P_N .

[0113] En JEM, para una CU con un tamaño menor o igual a 256 muestras de luminancia, una etiqueta a nivel CU se señala para indicar si se aplica o no OBMC a la CU actual. Para las CU con un tamaño mayor a 256 muestras de luminancia o no codificadas con modo de AMVP, OBMC se aplica por defecto. En el codificador,

cuando se aplica OBMC a una CU, su efecto se considera durante la etapa de estimación de movimiento. La señal de predicción con uso de información de movimiento del bloque vecino superior y el bloque vecino izquierdo se usa para compensar los límites superior e inferior de la señal original de la CU actual, y luego se aplica el proceso de estimación de movimiento normal.

[0114] Aunque BIO potencialmente proporciona más de 1 % de reducción de tasa de bits delta de Bjøntegaard (tasa BD) en JEM4.0, BIO también potencialmente introduce una complejidad computacional significativa y puede necesitar un aumento del ancho de banda de la memoria para el codificador y el decodificador. La presente divulgación describe técnicas que pueden reducir potencialmente la complejidad computacional y el ancho de banda de la memoria requerido en relación con BIO. Como un ejemplo, de acuerdo con las técnicas de la presente divulgación, un codificador de video puede determinar una cantidad de movimiento de BIO, por ejemplo, los valores v_x y v_y descritos con anterioridad, a nivel subbloque y usar dicha determinada cantidad de movimiento de BIO para modificar los valores de muestra de un bloque predictivo en función de cada muestra. En consecuencia, las técnicas de la presente divulgación pueden mejorar los codificadores de video y decodificadores de video al permitirles lograr las ventajas de codificación de BIO sin incurrir en una carga de la memoria y procesamiento sustancial necesaria para las implementaciones de BIO existentes.

[0115] En función de la ecuación (4), la presente divulgación introduce técnicas para reducir la complejidad de BIO mediante la redefinición de la ventana Ω . Tales técnicas pueden, por ejemplo, llevarse a cabo mediante el codificador de video 20 (por ejemplo, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44) o mediante el decodificador de video 30 (por ejemplo, la unidad de compensación de movimiento 72). La ventana Ω se define como cualquier bloque dentro del bloque actual que cubre el píxel actual con tamaño $M \times N$ en donde M y N son cualquier entero positivo. En un ejemplo, el bloque actual se divide en subbloques no superpuestos y la ventana Ω se define como el subbloque que cubre el píxel actual. En otro ejemplo, como se muestra en la Figura 11, el subbloque se define como el bloque más pequeño para el almacenamiento del vector de movimiento que cubre el píxel actual. En HEVC y JEM, el tamaño del bloque más pequeño es 4×4 . En otro ejemplo, el tamaño de la ventana Ω se adapta de acuerdo con la información de codificación, tal como el tamaño del bloque actual, los modos de codificación. Cuando el tamaño del bloque actual es mayor, se puede usar una ventana más grande Ω . Cuando el bloque actual se codifica como modos de subbloque, tales como fusión de

sub-CU, modo afín y modo FRUC, la ventana Ω se establece como el subbloque.

[0116] La Figura 11 muestra un ejemplo del BIO propuesto para un bloque 8x4, de acuerdo con las técnicas de la presente divulgación, con una ventana Ω para el píxel A, B y C. De acuerdo con las técnicas de la presente divulgación, pueden usarse ponderaciones equivalentes para resolver v_x y v_y como se muestra en la ecuación (7). En otro ejemplo, pueden usarse ponderaciones desiguales para resolver v_x y v_y como se muestra en la ecuación (10). Las ponderaciones desiguales pueden ser una función de las distancias entre el píxel central y los píxeles asociados. En aun otro ejemplo, la ponderación se puede calcular mediante un enfoque bilateral, por ejemplo, como se describe en https://en.wikipedia.org/wiki/Bilateral_filter. Además, se pueden usar tablas de consulta para almacenar todos los factores de ponderación para cada píxel de la ventana Ω en la ecuación (7).

[0117] En otro ejemplo, al derivar P_N para OBMC, el BIO solo se realiza para los píxeles parciales al derivar los predictores usando los movimientos vecinos. En un ejemplo, el BIO se inhabilita por completo para todos los píxeles en la derivación de P_N . En aun otro ejemplo, el BIO solo se aplica a los píxeles en las dos líneas externas como se muestra en las Figuras 12A-12D.

[0118] Además, para cada bloque, la cantidad de líneas en que se aplica BIO se puede señalar de manera explícita en el nivel del segmento de SPS/PPS. Si BIO se inhabilita, o se inhabilita de manera parcial, también se puede señalar explícitamente en el nivel del segmento de SPS/PPS.

[0119] En contraste, la cantidad de líneas en que se aplica BIO puede basarse implícitamente en ciertas condiciones de codificación, tales como modo CU (modo subbloque o modo no subbloque) o tamaño de bloque o la combinación de otras herramientas, tales como etiqueta de Compensación de Iluminación (IC) señalada. Si BIO se inhabilita, o se inhabilita de manera parcial, puede derivarse implícitamente en ciertas condiciones, tales como modo CU (modo subbloque o modo no subbloque) o tamaño de bloque o la combinación de otras herramientas, tales como etiqueta de IC señalada.

[0120] Las Figuras 12A-12D muestran ejemplos del BIO simplificado propuesto en OBMC de acuerdo con las técnicas de la presente divulgación, en donde x representa el predictor derivado sin BIO y y representa el predictor derivado con BIO. El refinamiento del vector de movimiento de BIO puede basarse en el bloque. Si el tamaño de bloque es M-por-N, una función de ponderación puede usarse para proporcionar diferentes factores de escala a píxeles de diferentes ubicaciones durante

el cálculo de los términos en la ecuación (7). Al resolver las ecuaciones (5) y (6), los píxeles interpolados y sus valores de gradiente deducidos del bloque entero pueden usarse para resolver vx y vy en conjunto, en lugar de resolver vx y vy de manera individual para cada posición del píxel.

[0121] En un ejemplo, el tamaño de ventana omega puede definirse como una ventana en ejecución centrada en cada ubicación de píxel, y se usa el valor promediado de la suma de los valores de todas las ubicaciones. Específicamente,

$$\begin{aligned}
 s_1 &= \sum_{k=0}^{N-1} \sum_{[i',j'] \in \Omega_k} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2 / N; s_3 = \sum_{k=0}^{N-1} \sum_{[i',j'] \in \Omega_k} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) / N; \\
 s_2 &= \sum_{k=0}^{N-1} \sum_{[i',j'] \in \Omega_k} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) / N; \\
 s_5 &= \sum_{k=0}^{N-1} \sum_{[i',j'] \in \Omega_k} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2 / N; s_6 = \sum_{k=0}^{N-1} \sum_{[i',j'] \in \Omega_k} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) / N
 \end{aligned} \tag{11}$$

en donde N es la cantidad de píxeles en cada subbloque, y Ω_k es la ventana definida para cada píxel. En un ejemplo, Ω_k puede ser la ventana 5x5 definida en el diseño de BIO actual para cada píxel, y así la función de ponderación puede determinarse con anticipación. Un ejemplo de función de ponderación usado para el subbloque 4x4 con la ventana 5x5 se muestra en la Figura 13. La Figura 13 muestra un ejemplo de una función de ponderación para un subbloque 4x4 con una ventana 5x5.

[0122] En otro ejemplo, la función de ponderación puede enviarse en SPS, PPS o encabezado de segmento. Para reducir los costos de señalización, un conjunto de funciones de ponderación predefinidas pueden almacenarse y solo los índices de las funciones de ponderación deben señalarse.

[0123] En otro ejemplo, el vector de movimiento refinado puede hallarse usando píxeles ubicados en la parte central del subbloque. Los valores de gradiente de píxeles centrales pueden calcularse usando un filtro de interpolación, y una ventana con tamaño M-por-N puede aplicarse a los píxeles interpolados para proporcionar diferentes ponderaciones a los píxeles centrales, a fin de calcular variables s1-s6 en la ecuación (7). En un ejemplo, se pueden calcular los valores de los gradientes de los puntos centrales, y se puede usar el valor promediado de los puntos centrales (ventana de ponderación equivalente). En otro ejemplo, un filtro de mediana puede usarse para seleccionar los píxeles representativos para calcular las variables s1-s6 en la ecuación (7).

[0124] En JVET-D0042, al resolver desviaciones de BIO, el tamaño de ventana para

cada píxel puede modificarse para ser el bloque actual completo, que potencialmente agrega complejidad computacional al diseño actual cuando un bloque actual es mayor o igual a 8×4 . El peor caso de las modificaciones es que una ventana 128×128 se use para la acumulación de gradientes y predictores para cada píxel en un bloque 128×128 .

[0125] Además, cuando los subbloques en una CU comparten el mismo MV o una CU intercodificada se divide en subbloques más pequeños en pos de la compensación de movimiento (MC), JEM-4.0 proporciona la flexibilidad para llevar a cabo la MC y el BIO para cada subbloque en paralelo, o realizar la MC y el BIO para el bloque más grande agregado o los subbloques con el mismo MV de una vez. De cualquier modo, JEM-4.0 proporciona resultados de codificación idénticos. Sin embargo, el BIO modificado en JVET-D0042 utiliza un cálculo de gradiente dependiente del tamaño de bloque y factores de ponderación, de modo que la realización de MC y BIO para dos bloques vecinos con el mismo movimiento en conjunto o por separado puede producir diferentes resultados. Para evitar los resultados diferentes, se debe especificar que el decodificador debe realizar MC y BIO a nivel de bloque o a cierto nivel subbloque. Dicha restricción puede ser demasiado estricta y no conveniente para implementación práctica de códec.

[0126] En función de la ecuación (4), la complejidad de BIO puede reducirse adicionalmente al redefinir la ventana Ω . Se definen dos tipos de la ventana Ω ; una es la ventana sin superposición y la otra es la ventana deslizante. Para la ventana de tipo sin superposición, el bloque actual se divide en subbloques sin superposición y la ventana Ω se define como el subbloque que cubre el píxel actual como se muestra en la Figura 11. Para la ventana de tipo deslizante, la ventana Ω se define como un bloque centrado en el píxel actual como se muestra en la Figura 7.

[0127] En ambos tipos de ventana Ω , el tamaño de la ventana Ω puede determinarse mediante diferentes métodos como se detalla a continuación. En adelante, se podrá suponer que la ventana Ω es un bloque rectangular de tamaño $M \times N$, en donde M y N puede ser cualquier entero negativo, tal como $(4 \times 4, 8 \times 8, 16 \times 16, 8 \times 4, \text{etcétera})$. La ventana Ω no se limita a una forma rectangular y puede tener cualquier otra forma, tal como forma de diamante. Las técnicas descritas también pueden aplicarse a formas distintas a la rectangular, si correspondiera.

[0128] El tamaño de la ventana puede ser fijo o variable, y puede ser predeterminado o señalizarse en la corriente de bits. Cuando el tamaño se señala, el tamaño puede señalizarse en el conjunto de parámetros de secuencias (SPS), conjunto de

parámetros de imágenes (PPS), encabezado de segmento, o a nivel de la CTU. El tamaño de ventana se puede determinar en conjunto mediante el tamaño del bloque de movimiento compensado (MC) por la ecuación indicada a continuación.

Tamaño de ventana horizontal $M = \min(M, MC_Size)$;

Tamaño de ventana vertical $N = \min(N, MC_Size)$.

[0129] En un ejemplo, el bloque de movimiento compensado (MC) depende solo de la información de codificación, tal como el tamaño del bloque actual y los modos de codificación. Por ejemplo, el bloque de movimiento compensado (MC) se establece como la CU completa cuando la CU actual se codifica con modos de no subbloque, tales como fusión de sub-CU, modo afín y modo FRUC. El bloque de movimiento compensado (MC) se establece como el subbloque cuando se usan los modos de subbloque, tales como fusión de sub-CU, modo afín y modo FRUC, independientemente de si subbloques tiene la misma información de movimiento.

[0130] En otro ejemplo, el bloque de movimiento compensado (MC) se define como el bloque de muestras en una CU que tiene el mismo MV. En este caso, el bloque de movimiento compensado (MC) se establece como la CU completa cuando la CU actual se codifica con modos de no subbloque, tales como fusión de sub-CU, modo afín y modo FRUC. Cuando una CU se codifica con modos de subbloque, tales como fusión de sub-CU, modo afín y modo FRUC, los subbloques con la misma información de movimiento se fusionan como un bloque de movimiento compensado (MC) con cierto orden de exploración de subbloque.

[0131] Tamaño adaptativo: el tamaño de la ventana Ω se adapta de acuerdo con la información de codificación, tal como el tamaño del bloque actual, los modos de codificación. En un ejemplo, la ventana Ω se establece como el bloque actual total o 1/4 del bloque actual cuando el bloque actual se codifica como modos de no subbloque, tales como fusión de sub-CU, modo afín y modo FRUC; y la ventana Ω se establece como subbloque cuando el bloque actual se codifica como modos de subbloque. El tamaño de ventana adaptativo puede determinarse en conjunto mediante el tamaño del bloque de movimiento compensado (MC) por la ecuación indicada a continuación.

Tamaño de ventana horizontal $M = \min(M, MC_Size)$;

Tamaño de ventana vertical $N = \min(N, MC_Size)$.

[0132] En las diversas técnicas para determinar el tamaño de la ventana Ω , una limitación de alto nivel del tamaño se puede incluir en una implementación de hardware o software fácil de usar. Por ejemplo, el tamaño de ventana debe ser menor

o igual al tamaño máximo de unidad de la transformada (TU) permitido en el sistema de códec de video. En otro ejemplo, el tamaño de ventana debe ser mayor o igual al bloque de MC más pequeño, tal como 4x4.

[0133] Para simplificar adicionalmente las operaciones en relación con BIO, la presente divulgación presenta técnicas para realizar el BIO como un posprocesamiento después de que toda predicción de movimiento compensado finalice por completo. Específicamente, después de realizar una MC convencional, OBMC puede aplicarse para generar mejores predictores para el bloque actual. En función del predictor final, BIO se aplica luego usando la información de movimiento del bloque actual para refinar adicionalmente el predictor. Por ejemplo, para el cálculo de gradientes en BIO, se puede usar el movimiento del bloque completo. En otro ejemplo, para cada subbloque, puede usarse el vector de movimiento promediado de OBMC. En otro ejemplo, para cada subbloque, puede usarse la mediana del vector de movimiento (para cada dimensión de manera individual).

[0134] Las funciones de ponderación pueden diseñarse de diferentes formas si se considera la derivación basada en el bloque de la refinación del vector de movimiento de BIO. Las ponderaciones equivalentes pueden usarse para cualquiera de los métodos mencionados con anterioridad. De manera alternativa, más ponderaciones pueden localizarse hacia la parte central de la ventana. En un ejemplo, las ponderaciones pueden calcularse mediante la distancia inversa (que incluye, entre otras, norma L1 o norma L2) entre el centro de la ventana y el píxel.

[0135] La Figura 14 es un diagrama de bloques que ilustra un ejemplo de codificador de video 20 que puede implementar técnicas para flujo óptico bidireccional. El codificador de video 20 puede realizar intracodificación e intercodificación de bloques de video dentro de segmentos de video. La intracodificación depende de la predicción espacial para reducir o eliminar la redundancia espacial en el video dentro de un cuadro de video o imagen determinados. La intercodificación depende de la predicción temporal para reducir o eliminar la redundancia temporal en el video dentro de cuadros o imágenes adyacentes de una secuencia de video. El intramodo (modo I) puede hacer referencia a cualquiera de los diversos modos de codificación en función del espacio. Los intermodos, tales como predicción unidireccional (modo P) o bipredicción (modo B), pueden hacer referencia a cualquiera de los diversos modos de codificación en función del tiempo.

[0136] Como se muestra en la Figura 14, el codificador de video 20 recibe datos de video y almacena los datos de video recibidos en la memoria de datos de video 38. La

memoria de datos de video 38 puede almacenar los datos de video que se desea codificar mediante los componentes del codificador de video 20. Los datos de video almacenados en la memoria de datos de video 38 se pueden obtener, por ejemplo, de la fuente de video 18. La memoria de imágenes de referencia 64 puede ser una memoria de imágenes de referencia que almacena datos de video de referencia para usar en la codificación de datos de video mediante el codificador de video 20, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de video 38 y la memoria de imágenes de referencia 64 se pueden formar con una variedad de dispositivos de memoria, tal como memoria dinámica de acceso aleatorio (DRAM), que incluye DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de video 38 y la memoria de imágenes de referencia 64 se pueden proporcionar mediante el mismo dispositivo de memoria o mediante dispositivos de memoria diferentes. En diversos ejemplos, la memoria de datos de video 38 puede ser interna (*on-chip*) con otros componentes del codificador de video 20 o externa (*off-chip*) con respecto a aquellos componentes.

[0137] El codificador de video 20 recibe un bloque de video actual dentro de un cuadro de video que se desea codificar. En el ejemplo de la Figura 14, el codificador de video 20 incluye una unidad de selección de modo 40, memoria de imágenes de referencia 64 (que también puede denominarse memoria de imagen decodificada (DPB)), sumador 50, unidad de procesamiento de la transformada 52, unidad de cuantificación 54 y unidad de codificación entrópica 56. A su vez, la unidad de selección de modo 40 incluye una unidad de compensación de movimiento 44, unidad de estimación de movimiento 42, unidad de procesamiento de intrapredicción 46 y unidad de partición 48. Para la reconstrucción del bloque de video, el codificador de video 20 también incluye unidad de cuantificación inversa 58, unidad de procesamiento de la transformada inversa 60 y sumador 62. Un filtro de eliminación de bloques (no se muestra en la Figura 14) también se puede incluir para filtrar los límites del bloque para eliminar artefactos de compresión del video reconstruido. Si se desea, el filtro de eliminación de bloques filtraría, en general, la salida del sumador 62. También se pueden usar filtros adicionales (en bucle o posbucle) además del filtro de eliminación de bloques. Tales filtros no se muestran para mayor brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

[0138] Durante el proceso de codificación, el codificador de video 20 recibe un cuadro o un segmento de video que se desean codificar. El cuadro o el segmento se pueden

dividir en múltiples bloques de video. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación interpredictiva del bloque de video recibido con respecto a uno o más bloques en uno o más cuadros de referencia para proporcionar predicción temporal. La unidad de procesamiento de intrapredicción 46 puede, de manera alternativa, intraprededir el bloque de video recibido usando píxeles de uno o más bloques vecinos en el mismo cuadro o segmento como el bloque que se desea codificar para proporcionar predicción espacial. El codificador de video 20 puede realizar múltiples pases de codificación, por ejemplo, para seleccionar un modo de codificación adecuado para cada bloque de datos de video.

[0139] Asimismo, la unidad de partición 48 puede particionar bloques de datos de video en subbloques, en función de la evaluación de los esquemas de partición previos en pases de codificación previos. Por ejemplo, la unidad de partición 48 puede particionar inicialmente un cuadro o un segmento en LCU, y particionar cada una de las LCU en sub-CU en función del análisis de tasa-distorsión (por ejemplo, optimización de tasa-distorsión). La unidad de selección de modo 40 puede producir adicionalmente una estructura de datos de árbol cuaternario que indica la partición de una LCU en sub-CU. Las CU con nodo hoja del árbol cuaternario pueden incluir una o más PU y una o más TU.

[0140] La unidad de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, en función de los resultados de errores, y proporcionar el bloque previsto resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para usarlo como un cuadro de referencia. La unidad de selección de modo 40 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores intramodo, información de partición y otra información sintáctica, a la unidad de codificación entrópica 56.

[0141] La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado para fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman movimiento para los bloques de video. Por ejemplo, un vector de movimiento, puede indicar el desplazamiento de una PU de un bloque de video dentro de un cuadro de video o una imagen actuales con respecto a un bloque predictivo dentro de un cuadro de referencia (u otra unidad codificada) con respecto al bloque actual que se codifica dentro del cuadro actual (u otra unidad codificada). Un bloque

predictivo es un bloque que coincide en gran medida con el bloque que se desea codificar, en cuanto a diferencia de píxeles, que se puede determinar mediante la suma de la diferencia absoluta (SAD), suma de diferencia al cuadrado (SSD) u otras medidas de diferencia. En algunos ejemplos, el codificador de video 20 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de video 20 puede interpolar valores de posiciones de 1/4 píxel, posiciones de 1/8 píxel u otras posiciones de píxeles fraccionarios de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y posiciones de píxeles fraccionarios y extraer un vector de movimiento con precisión de píxeles fraccionarios.

[0142] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de video en un segmento intercodificado mediante la comparación de la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación entrópica 56 y a la unidad de compensación de movimiento 44.

[0143] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede incluir buscar o generar el bloque predictivo en función del vector de movimiento determinado por la unidad de estimación de movimiento 42. Nuevamente, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar funcionalmente integradas, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de video actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que el vector de movimiento señala en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de video residual restando valores de píxeles del bloque predictivo de los valores de píxeles del bloque de video actual que se codifica, lo que genera valores de diferencia de píxeles, como se indica más adelante. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento con respecto a componentes de luminancia, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en función de los componentes de luminancia para

los componentes de crominancia y componentes de luminancia. La unidad de selección de modo 40 también puede producir elementos sintácticos asociados a los bloques de video y el segmento de video para usar con el decodificador de video 30 en la decodificación de los bloques de video del segmento de video.

[0144] Asimismo, la unidad de compensación de movimiento 44 se puede configurar para llevar a cabo cualquiera o la totalidad de las técnicas de la presente divulgación (solas o combinadas). Aunque se analizó con respecto a la unidad de compensación de movimiento 44, cabe destacar que la unidad de selección de modo 40, unidad de estimación de movimiento 42, unidad de partición 48 y/o unidad de codificación entrópica 56 también pueden configurarse para llevar a cabo ciertas técnicas de la presente divulgación, solas o en combinación con la unidad de compensación de movimiento 44. En un ejemplo, la unidad de compensación de movimiento 44 se puede configurar para llevar a cabo las técnicas de BIO analizadas en la presente.

[0145] La unidad de procesamiento de intrapredicción 46 puede intrapredecir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se describió anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación diferentes, y la unidad de procesamiento de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción adecuado para usar de los modos evaluados.

[0146] Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular los valores de tasa-distorsión usando un análisis de tasa-distorsión para diversos modos de intrapredicción evaluados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa-distorsión entre los modos evaluados. En general, el análisis de tasa-distorsión determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una tasa de bits (es decir, una cantidad de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular relaciones de las distorsiones y las tasas para diversos bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa-distorsión para el bloque.

[0147] Después de seleccionar un modo de intrapredicción para un bloque, la unidad

de procesamiento de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación entrópica 56. La unidad de codificación entrópica 56 puede codificar la información que indica el modo de intrapredicción seleccionado. El codificador de video 20 puede incluir en los datos de la configuración de la corriente de bits transmitidos, que pueden incluir una pluralidad de tablas indexadas de modo de intrapredicción y una pluralidad de tablas indexadas de modo de intrapredicción modificado (también denominadas tablas de mapeo de codeword), definiciones de contextos de codificación para diversos bloques e indicaciones del modo de intrapredicción más probable, una tabla indexada de modo de intrapredicción y una tabla indexada de modo de intrapredicción modificado para usar en cada uno de los contextos.

[0148] El codificador de video 20 forma un bloque de video residual restando los datos de predicción de la unidad de selección de modo 40 del bloque de video original que se codifica. El sumador 50 representa el componente o los componentes que realizan esta operación de sustracción. La unidad de procesamiento de la transformada 52 aplica una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar, al bloque residual, lo que produce un bloque de video que comprende los valores del coeficiente de la transformada. Se pueden usar transformadas *wavelet*, transformadas enteras, transformadas de subbanda, transformadas discretas del seno (DST) u otros tipos de transformadas en lugar de una DCT. En cualquier caso, la unidad de procesamiento de la transformada 52 aplica la transformada al bloque residual, lo que produce un bloque de coeficientes de la transformada. La transformada puede convertir la información residual de un dominio de píxel en un dominio de la transformada, tal como un dominio de frecuencia. La unidad de procesamiento de la transformada 52 puede enviar los coeficientes de la transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de la transformada para reducir adicionalmente la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación.

[0149] Luego de la cuantificación, la unidad de codificación entrópica 56 codifica por entropía los coeficientes de la transformada cuantificados. Por ejemplo, la unidad de codificación entrópica 56 puede realizar codificación adaptativa según el contexto de longitud variable (CAVLC), codificación aritmética binaria adaptativa según el contexto (CABAC), codificación aritmética binaria adaptativa según el contexto en función de la

sintaxis (SBAC), codificación entrópica de partición en intervalos de probabilidad (PIPE) u otra técnica de codificación entrópica. En el caso de codificación entrópica según el contexto, el contexto puede ser en función de bloques vecinos. Luego de la codificación entrópica mediante la unidad de codificación entrópica 56, la corriente de bits codificada se puede transmitir a otro dispositivo (por ejemplo, el decodificador de video 30) o archivar para la posterior transmisión o recuperación.

[0150] La unidad de cuantificación inversa 58 y la unidad de procesamiento de la transformada inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles. En particular, el sumador 62 agrega el bloque residual reconstruido al bloque de predicción de movimiento compensado que se produjo anteriormente mediante la unidad de compensación de movimiento 44 o unidad de procesamiento de intrapredicción 46 para producir un bloque de video reconstruido para almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar el bloque de video reconstruido como un bloque de referencia para intercodificar un bloque en un cuadro de video posterior.

[0151] La Figura 15 es un diagrama de bloques que ilustra un ejemplo de decodificador de video 30 que puede implementar técnicas para flujo óptico bidireccional. En el ejemplo de la Figura 15, el decodificador de video 30 incluye una unidad de decodificación entrópica 70, unidad de compensación de movimiento 72, unidad de procesamiento de intrapredicción 74, unidad de cuantificación inversa 76, unidad de procesamiento de la transformada inversa 78, memoria de imágenes de referencia 82 y sumador 80. El decodificador de video 30 puede, en algunos ejemplos, realizar un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador de video 20 (Figura 14). La unidad de compensación de movimiento 72 puede generar datos de predicción en función de vectores de movimiento recibidos de la unidad de codificación entrópica 70, mientras que la unidad de procesamiento de intrapredicción 74 puede generar datos de predicción en función de los indicadores de modo de intrapredicción recibidos de la unidad de decodificación entrópica 70.

[0152] Durante el proceso de decodificación, el decodificador de video 30 recibe una corriente de bits del video codificado que representa bloques de video de un segmento de video codificado y elementos sintácticos asociados del codificador de video 20. El decodificador de video 30 almacena la corriente de bits del video codificado recibida

en la memoria de datos de video 68. La memoria de datos de video 68 puede almacenar datos de video, tales como una corriente de bits de video codificado, a ser decodificados por los componentes del decodificador de video 30. Los datos de video almacenados en la memoria de datos de video 68 se pueden obtener, por ejemplo, mediante un medio legible por computadora 16, de un medio de almacenamiento, o de una fuente de video local, tal como una cámara, o mediante el acceso a un medio de almacenamiento de datos físicos. La memoria de datos de video 85 puede formar una memoria de imagen codificada (CPB) que almacena datos de video codificados de una corriente de bits de video codificada. La memoria de imágenes de referencia 82 puede ser una memoria de imágenes de referencia que almacena datos de video de referencia para usar en la decodificación de datos de video mediante el decodificador de video 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de video 68 y la memoria de imágenes de referencia 82 se pueden formar con una variedad de dispositivos de memoria, tal como DRAM, SDRAM, MRAM, RRAM u otros tipos de dispositivos de memoria. La memoria de datos de video 68 y la memoria de imágenes de referencia 82 se pueden proporcionar mediante el mismo dispositivo de memoria o mediante dispositivos de memoria diferentes. En diversos ejemplos, la memoria de datos de video 68 puede ser interna con otros componentes del decodificador de video 30 o externa con respecto a aquellos componentes.

[0153] Durante el proceso de decodificación, el decodificador de video 30 recibe una corriente de bits del video codificado que representa bloques de video de un segmento de video codificado y elementos sintácticos asociados del codificador de video 20. La unidad de decodificación entrópica 70 del decodificador de video 30 decodifica por entropía la corriente de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción, y otros elementos sintácticos. La unidad de decodificación entrópica 70 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de video 30 puede recibir los elementos sintácticos al nivel del segmento de video y/o al nivel del bloque de video.

[0154] Cuando el segmento de video se codifica como un segmento intracodificado (I), la unidad de procesamiento de intrapredicción 74 puede generar datos de predicción para un bloque de video del segmento de video actual en función de un modo de intrapredicción señalado y datos de los bloques decodificados previamente del cuadro o la imagen actuales. Cuando el cuadro de video se codifica como un

segmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 produce bloques predictivos para un bloque de video del segmento de video actual en función de los vectores de movimiento y otros elementos sintácticos recibidos de la unidad de decodificación entrópica 70. Los bloques predictivos se pueden producir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de video 30 puede construir las listas de cuadros de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas en función de las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

[0155] La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de video del segmento de video actual mediante el análisis de los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir bloques predictivos para el bloque de video actual que se decodifica. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intra- o interpredicción) usado para codificar los bloques de video del segmento de video, un tipo de segmento de interpredicción (por ejemplo, segmento B, segmento P o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de video intercodificado del segmento, estado de interpredicción para cada bloque de video intercodificado del segmento y otra información para decodificar los bloques de video en el segmento de video actual.

[0156] La unidad de compensación de movimiento 72 también puede realizar interpolación en función de filtros de interpolación para precisión subpíxel. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los utiliza el codificador de video 20 durante la codificación de los bloques de video para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de video 20 de los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

[0157] Asimismo, la unidad de compensación de movimiento 72 se puede configurar para llevar a cabo cualquiera o la totalidad de las técnicas de la presente divulgación (solas o combinadas). Por ejemplo, la unidad de compensación de movimiento 72 se puede configurar para llevar a cabo las técnicas de BIO analizadas en la presente.

[0158] La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir,

descuantifica, los coeficientes de la transformada cuantificados proporcionados en la corriente de bits y decodificados por la unidad de decodificación entrópica 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y calculado por el decodificador de video 30 para cada bloque de video en el segmento de video para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar.

[0159] La unidad de procesamiento de la transformada inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa o un proceso de la transformada inversa conceptualmente similar, a los coeficientes de la transformada con el fin de producir bloques residuales en el dominio de píxeles.

[0160] Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de video actual en función de los vectores de movimiento y otros elementos sintácticos, el decodificador de video 30 forma un bloque de video decodificado sumando los bloques residuales de la unidad de procesamiento de la transformada inversa 78 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, un filtro de eliminación de bloques también se puede aplicar para filtrar los bloques decodificados con el fin de eliminar los artefactos de compresión. Otros filtros en bucle (ya sea en el bucle de codificación o después del bucle de codificación) también se pueden usar para suavizar las transiciones de píxeles o, de otro modo, mejorar la calidad de video. Los bloques de video decodificados en un cuadro o imagen determinados luego se almacenan en la memoria de imágenes de referencia 82, que almacena imágenes de referencia utilizadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 también almacena videos decodificados para la posterior presentación en un dispositivo de visualización, tal como un dispositivo de visualización 32 de la Figura 1. Por ejemplo, la memoria de imágenes de referencia 82 puede almacenar imágenes decodificadas.

[0161] La Figura 16 es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de video, para decodificar datos de video, de acuerdo con una técnica de la presente divulgación. El decodificador de video descrito con respecto a la Figura 16 puede, por ejemplo, ser un decodificador de video, tal como un decodificador de video 30, para proporcionar un video decodificado visualizable o puede ser un decodificador de video implementado en un codificador de video, tal como el bucle de decodificación del codificador de video 20, que incluye una unidad de cuantificación

inversa 58, unidad de procesamiento de la transformada inversa 60, sumador 62, y memoria de imágenes de referencia 64, así como también porciones de la unidad de selección de modo 40.

[0162] De acuerdo con las técnicas de la Figura 16, el decodificador de video determina un bloque de datos de video que se codifica mediante un modo de interpredicción bidireccional (200). El decodificador de video determina un primer vector de movimiento para el bloque que apunta a una primera imagen de referencia (202). El decodificador de video determina un segundo MV para el bloque que apunta a una segunda imagen de referencia, en donde la primera imagen de referencia es diferente de la segunda imagen de referencia (204). El decodificador de video usa el primer MV para localizar un primer bloque predictivo en la primera imagen de referencia (206). El decodificador de video usa el segundo MV para localizar un segundo bloque predictivo en la segunda imagen de referencia (208).

[0163] El decodificador de video determina una primera cantidad de movimiento de BIO para un primer subbloque del primer bloque predictivo (210). El primer subbloque puede ser diferente de una unidad de codificación, una unidad de predicción, y una unidad de la transformada para el bloque. A fin de determinar la primera cantidad de movimiento de BIO, el decodificador de video puede, en algunos ejemplos, determinar la primera cantidad de movimiento de BIO en función de las muestras en el primer subbloque y las muestras fuera del primer subbloque, y en otros ejemplos, determinar la primera cantidad de movimiento de BIO solamente en función de las muestras en el primer subbloque. La primera cantidad de movimiento de BIO puede incluir, por ejemplo, un campo de vector de movimiento que incluye un componente horizontal y un componente vertical.

[0164] El decodificador de video determina un primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, un primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO (212). A fin de determinar el primer subbloque predictivo final para el bloque de datos de video en función del primer subbloque del primer bloque predictivo, el primer subbloque del segundo bloque predictivo, y la primera cantidad de movimiento de BIO, el decodificador de video puede determinar el primer subbloque predictivo final usando, por ejemplo, la ecuación (2) expresada más arriba.

[0165] El decodificador de video determina una segunda cantidad de movimiento de BIO para un segundo subbloque del primer bloque predictivo (214). El segundo subbloque puede ser diferente de una unidad de codificación, una unidad de

predicción, y una unidad de la transformada para el bloque. A fin de determinar la segunda cantidad de movimiento de BIO, el decodificador de video puede, en algunos ejemplos, determinar la segunda cantidad de movimiento de BIO en función de las muestras en el segundo subbloque y las muestras fuera del segundo subbloque, y en otro ejemplo, determinar la segunda cantidad de movimiento de BIO solamente en función de las muestras en el segundo subbloque. La segunda cantidad de movimiento de BIO puede incluir, por ejemplo, un campo de vector de movimiento que incluye un componente horizontal y un componente vertical.

[0166] El decodificador de video determina un segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO (216). A fin de determinar el segundo subbloque predictivo final para el bloque de datos de video en función del segundo subbloque del primer bloque predictivo, un segundo subbloque del segundo bloque predictivo, y la segunda cantidad de movimiento de BIO, el decodificador de video puede determinar, por ejemplo, el segundo subbloque predictivo final usando, por ejemplo, la ecuación (2).

[0167] El decodificador de video determina un bloque predictivo final para el bloque de datos de video en función del primer subbloque predictivo final y el segundo subbloque predictivo final (218). El decodificador de video puede agregar, por ejemplo, datos residuales al bloque predictivo final para determinar un bloque reconstruido para el bloque de datos de video. El decodificador de video puede llevar a cabo también uno o más procesos de filtrado en el bloque reconstruido de datos de video.

[0168] El decodificador de video proporciona una imagen de datos de video que comprende una versión decodificada del bloque de datos de video (220). Si la decodificación se lleva a cabo como parte de un bucle de decodificación de un proceso de codificación de video, entonces el decodificador de video puede proporcionar, por ejemplo, la imagen al almacenar la imagen en una memoria de imágenes de referencia, y el decodificador de video puede usar la imagen como una imagen de referencia al codificar otra imagen de los datos de video. Si el decodificador de video es un decodificador de video configurado para proporcionar un video decodificado visualizable, entonces el decodificador de video puede proporcionar, por ejemplo, la imagen de datos de video a un dispositivo de visualización.

[0169] Cabe aclarar que según el ejemplo, algunos actos o eventos de cualquiera de las técnicas descritas en la presente se pueden realizar en una secuencia diferente, se

pueden agregar, fusionar o excluir completamente (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Asimismo, en algunos ejemplos, los actos o eventos se pueden realizar de manera concurrente, por ejemplo, a través de procesamiento multihilo, procesamiento interrumpido o procesadores múltiples, en lugar de secuencialmente.

[0170] 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 código en un medio legible por computadora y pueden ejecutarse mediante una unidad de procesamiento de hardware. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, lo que corresponde a un medio tangible, tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa de computadora desde 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) un medio de almacenamiento tangible legible por computadora que no sea transitorio 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 se puede acceder mediante una o más computadoras o uno o más procesadores para obtener instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en la presente descripción. Un producto de programa de computación puede incluir un medio legible por computadora.

[0171] A modo ejemplificativo, y no limitativo, tal medio de almacenamiento legible por computadora puede comprender RAM, ROM, EEPROM, CD-ROM u otros dispositivos de almacenamiento en disco óptico, de almacenamiento en disco magnético u otro almacenamiento magnético, o cualquier otro medio que pueda utilizarse para almacenar el código de programación deseado en forma de instrucciones o estructuras de datos y al que puede accederse por medio de una computadora. Además, cualquier conexión se conoce con el término medio legible por computadora. Por ejemplo, si las instrucciones son transmitidas desde una página web, servidor u otra fuente remota que utilice un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL), o tecnologías inalámbricas tales como infrarrojo, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL, o tecnologías inalámbricas tales como infrarrojo, radio, y microondas se incluyen en la definición de

medio. Sin embargo, cabe aclarar que los medios de almacenamiento legibles por computadora no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en su lugar, son dirigidos a medios de almacenamiento tangibles no transitorios. Disquete y disco, como se usa en la presente, incluye disco compacto (CD), disco láser, disco óptico, disco digital versátil (DVD), disco flexible y disco Blu-ray, en donde los disquetes reproducen datos en forma magnética, mientras que los discos reproducen datos en forma óptica con láser. También deberían incluirse combinaciones de lo anterior dentro del alcance de medios legibles por computadora.

[0172] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más DSP, procesadores de propósito general, ASIC, FPGA u otro circuito lógico integrado o separado equivalente. En consecuencia, como se usa en la presente, el término “procesador” puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente. Además, en algunos aspectos, la funcionalidad descrita en la presente puede estar proporcionada dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o puede estar incorporada en un códec combinado. Además, las técnicas podrían implementarse totalmente en uno o más circuitos o elementos lógicos.

[0173] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un microteléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen distintos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no es necesario que sean realizadas por diferentes unidades de hardware. En su lugar, como se describió anteriormente, distintas unidades pueden combinarse en una unidad de hardware de códec o pueden ser proporcionadas por un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores, como se describió anteriormente, junto con software y/o firmware adecuado.

[0174] Se han descrito distintos ejemplos. Estos y otros ejemplos se encuentran dentro del alcance de las siguientes reivindicaciones.