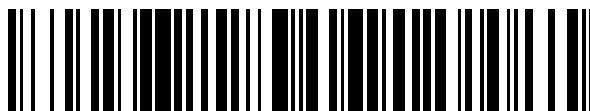


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 779 461**

51 Int. Cl.:

H04N 19/593 (2014.01)
H04N 19/176 (2014.01)
H04N 19/11 (2014.01)
H04N 19/117 (2014.01)
H04N 19/146 (2014.01)
H04N 19/182 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.09.2016 PCT/US2016/053126**

87 Fecha y número de publicación internacional: **06.04.2017 WO17058635**

96 Fecha de presentación y número de la solicitud europea: **22.09.2016 E 16775947 (1)**

97 Fecha y número de publicación de la concesión europea: **18.12.2019 EP 3357247**

54 Título: **Intrapredicción de vídeo mejorada que usa una combinación de predicción dependiente de la posición para codificación de vídeo**

30 Prioridad:

29.09.2015 US 201562234645 P
21.09.2016 US 201615272034

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
17.08.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

SAID, AMIR;
ZHAO, XIN;
CHEN, JIANLE y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 779 461 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Intrapredicción de vídeo mejorada que usa una combinación de predicción dependiente de la posición para codificación de vídeo

5 **[0001]** Esta solicitud reivindica el beneficio de la solicitud de patente provisional de EE. UU. n.º 62/234,645 presentada el 29 de septiembre de 2015.

CAMPO TÉCNICO

10 **[0002]** Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

15 **[0003]** Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radioteléfonos celulares o por satélite, los llamados "teléfonos inteligentes", dispositivos de teleconferencia por vídeo, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en diversas normas definidas por 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, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC) e ITU-T H.265, Codificación de vídeo de alta eficacia (HEVC) y ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente implementando dichas técnicas de codificación de vídeo.

20 **[0004]** Las técnicas de codificación de vídeo incluyen predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un sector de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un sector intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un sector intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a unas muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a unas muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

30 **[0005]** La predicción espacial o temporal dan como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a 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 con los datos residuales. Para una compresión adicional, los datos residuales se pueden transformar desde el dominio del píxel a un dominio de la transformada, dando como resultado unos coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación entrópica para conseguir todavía más compresión.

SUMARIO

40 **[0006]** En general, esta divulgación describe técnicas relacionadas con una intrapredicción de vídeo mejorada que usa una combinación de predicción dependiente de la posición en la codificación de vídeo. Las técnicas se pueden usar en el contexto de los códecs de vídeo avanzados, tales como de las ampliaciones de HEVC o la próxima generación de normas de codificación de vídeo. En HEVC, por ejemplo, se usa un conjunto de 35 predictores lineales para realizar intracodificación y la predicción se puede calcular a partir de un conjunto no filtrado o filtrado de píxeles de "referencia" vecinos, dependiendo del modo de predictor seleccionado y el tamaño de bloque. Las técnicas de esta divulgación pueden usar una combinación ponderada de los conjuntos no filtrados y filtrados de píxeles de referencia para lograr una mejor compresión (por medio de, por ejemplo, una predicción mejorada y, por lo tanto, un pequeño residuo), permitir un cálculo paralelo eficaz de todos los conjuntos de valores de predicción y mantener baja la complejidad (por ejemplo, aplicando filtrado solo a un conjunto de píxeles de referencia y no a los propios valores predichos).

50 **[0007]** En un ejemplo, esta divulgación está dirigida a un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento descodificar bloques vecinos de un bloque actual en una imagen de datos de vídeo; formar una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados que

comprenden versiones filtradas de píxeles vecinos del bloque actual en los bloques vecinos; formar una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados correspondientes a versiones no filtradas de los píxeles vecinos; calcular un conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual en base a una combinación lineal de uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada multiplicada por un primer conjunto de pesos y uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada multiplicada por un segundo conjunto de pesos; y reconstruir el bloque actual en base al conjunto calculado de valores predictivos.

[0008] En otro ejemplo, esta divulgación está dirigida a un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento recibir un bloque actual de una imagen, una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados y una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados; descodificar bloques vecinos de un bloque actual en una imagen de datos de vídeo; formar una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados que comprenden versiones filtradas de píxeles vecinos del bloque actual en los bloques vecinos; formar una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados correspondientes a versiones no filtradas de los píxeles vecinos; generar un bloque predictivo para el bloque actual, en el que generar comprende calcular un conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual en base a una combinación lineal de uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada multiplicada por un primer conjunto de pesos y uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada multiplicada por un segundo conjunto de pesos; generar un bloque residual en base a una diferencia entre el bloque actual y el bloque predictivo; y codificar datos que representan el bloque residual en un flujo de bits.

[0009] En otro ejemplo, esta divulgación está dirigida a un dispositivo para descodificar datos de vídeo, comprendiendo el dispositivo una memoria y uno o más procesadores en comunicación con la memoria. El uno o más procesadores están configurados para descodificar bloques vecinos de un bloque actual en una imagen de datos de vídeo; formar una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados que comprenden versiones filtradas de píxeles vecinos del bloque actual en los bloques vecinos; formar una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados correspondientes a versiones no filtradas de los píxeles vecinos; calcular un conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual en base a una combinación lineal de uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada multiplicada por un primer conjunto de pesos y uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada multiplicada por un segundo conjunto de pesos; y reconstruir el bloque actual en base al conjunto calculado de valores predictivos.

[0010] En otro ejemplo, esta divulgación está dirigida a un dispositivo para codificar datos de vídeo como se define en la reivindicación 14.

[0011] En otro ejemplo, esta divulgación está dirigida a un dispositivo para descodificar datos de vídeo como se define en la reivindicación 13.

[0012] En otro ejemplo, esta divulgación está dirigida a un medio no transitorio legible por ordenador como se define en la reivindicación 15.

[0013] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y los dibujos y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0014]

La FIG. 1 es un diagrama que ilustra un bloque de 4 x 4 píxeles que se van a predecir en la codificación intratrama que puede utilizar unas técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama que ilustra unas configuraciones de filtro para los predictores direccionales en base al tamaño de bloque.

La FIG. 3 ilustra una predicción de un bloque 4x4 usando una referencia sin filtrar de acuerdo con unas técnicas de la presente divulgación.

La FIG. 4 ilustra una predicción de un bloque 4x4 usando una referencia filtrada de acuerdo con unas técnicas de la presente divulgación.

La FIG. 5 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar unas técnicas descritas en esta divulgación.

La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar unas técnicas descritas en esta divulgación.

5 La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar unas técnicas descritas en esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo de acuerdo con las técnicas de esta divulgación.

10 La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de ejemplo de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

15 **[0015]** En general, esta divulgación describe técnicas relacionadas con una intrapredicción de vídeo mejorada que usa una combinación de predicción dependiente de la posición en la codificación de vídeo. Las técnicas se pueden usar en el contexto de los códecs de vídeo avanzados, tales como de las ampliaciones de HEVC o la próxima generación de normas de codificación de vídeo.

20 **[0016]** Las técnicas de esta divulgación se describen en general con respecto a ITU-T H.265, también conocida como Codificación de vídeo de alta eficacia (HEVC), que se describe en el documento "SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services-Coding of moving video" Codificación de vídeo de alta eficacia, ITU-T H.265, abril de 2013. Sin embargo, estas técnicas se pueden aplicar a otras normas de codificación de vídeo, que incluyen ampliaciones de HEVC y ampliaciones de otras normas. Los ejemplos de otras normas de codificación de vídeo 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 e ITU -TH.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), incluyendo sus ampliaciones Codificación de vídeo escalable (SVC) y Codificación de vídeo multivista (MVC).

30 **[0017]** El Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) ISO/IEC finalizaron recientemente la norma H.265. El último borrador de especificación HEVC, a continuación en el presente documento denominado HEVC WD, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip.

35 **[0018]** El JCT-3V también ha elaborado la ampliación multivista de HEVC, MV-HEVC. Un ejemplo de borrador de trabajo (WD) de MV-HEVC, denominado MV-HEVC WD8, está disponible en phenix.it-sudparis.eu/jct2/doc_end_user/documents/8_Valencia/wg11/JCT3V-H1002-v5.zip. El JCT-VC también ha elaborado una ampliación escalable de HEVC, denominada SHVC. Un ejemplo de borrador de trabajo (WD) de SHVC, denominado SHVC WD6, está disponible en phenix.it-sudparis.eu/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1008-v2.zip.

45 **[0019]** En la norma de compresión de vídeo HEVC, la predicción intratrama se aplica a bloques con $N \times N$ píxeles, usando un grupo de $4N + 1$ píxeles que son adyacentes al bloque que se va a predecir, a saber, unos *píxeles de referencia*. Esos *píxeles de referencia* tienen valores que se han reconstruido y son conocidos en el codificador y el descodificador cuando se calcula la predicción de bloque. Por ejemplo, la FIG. 1 muestra un bloque de 4×4 píxeles que se van a predecir en el procedimiento de codificación intratrama usado por la HEVC. Los píxeles que se van a predecir 90 se muestran como cuadrados blancos y los píxeles de referencia 92 como cuadrados grises.

50 **[0020]** La intrapredicción es una herramienta de codificación que se emplea para reducir o eliminar redundancias espaciales usando píxeles vecinos dentro de una imagen. Para encontrar direcciones de predicción exactas y eliminar redundancias eficazmente, la HEVC puede usar hasta 35 modos de predicción para cada PU. Puede haber dos conjuntos de valores que se usan para la predicción, dependiendo del modo de predictor (plano, DC o direccional) y el tamaño de bloque. Para cada modo, se puede usar una versión no filtrada o filtrada de los píxeles de referencia para la predicción. La HEVC, por ejemplo, define una tabla fija para determinar si se han de usar píxeles de referencia filtrados o no filtrados en la intrapredicción. El documento WO2012/102929 divulga la intrapredicción usando píxeles de referencia filtrados y no filtrados. Cada píxel de predicción usa valores de referencia filtrados o no filtrados de acuerdo con unos datos estadísticos locales del píxel que se va a codificar o su vecindad causal. La selección también puede depender de la posición, seleccionando los píxeles más cercanos a los vecinos causales la referencia no filtrada.

60 **[0021]** La FIG. 2 es un diagrama conceptual que ilustra una representación gráfica que define configuraciones de filtro para predicción direccional en base al tamaño de bloque. En la FIG. 2, se muestra un círculo donde un modo de predicción (por ejemplo, el modo de predicción direccional) usa un filtrado de muestras de referencia para un tamaño de bloque en particular (por ejemplo, 8×8 , 16×16 y 32×32). En la FIG. 2, se muestra una "x" gris donde un modo de predicción usa muestras de referencia no filtradas para un tamaño de bloque en particular. Por ejemplo, los modos de predicción 2, 18 y 34 pueden usar píxeles de referencia filtrados independientemente del tamaño de bloque, los modos

10 y 26 pueden utilizar píxeles de referencia no filtrados independientemente del tamaño de bloque, los modos 3-8, 12-17, 19-24 y 28-33 pueden utilizar píxeles de referencia no filtrados en bloques de tamaño 8x8 pero usan píxeles de referencia filtrados en bloques de tamaños 16x16 y 32x32, y los modos 9, 11, 25 y 27 utilizan píxeles de referencia no filtrados en bloques de tamaños 8x8 y 16x16 y píxeles de referencia filtrados en bloques de tamaño 32x32. Además, puede haber un tipo de filtro de paso bajo que se puede usar, con las tres tomas (1/4, 1/2, 1/4).

[0022] Los procedimientos actuales pueden ser más adecuados para los vídeos de baja resolución usados en el pasado, donde la mejor compresión se logra principalmente con bloques pequeños (por ejemplo, 4 x 4 u 8 x 8). Sin embargo, una tendencia más reciente es tener vídeo 1080 x 1920 HD o vídeo 2160 x 3840 UHD. Para esas resoluciones, la predicción usando tamaños de bloque mayores puede ser más eficaz y obtener una mejor compresión.

[0023] Se puede utilizar un filtrado de paso bajo más fuerte de la referencia en esos bloques más grandes para obtener una mejor compresión. Sin embargo, en esos casos, un único filtro de paso bajo que optimiza la predicción para todos los bloques de un tamaño de bloque determinado y el filtro óptimo varían de acuerdo con la textura en cada bloque.

[0024] Un dispositivo de vídeo que implementa la norma HEVC actual usa una predicción que está basada en la referencia filtrada o en la referencia no filtrada, que no admite una forma de combinar información de esos dos casos, aunque esos datos pueden estar fácilmente disponibles en el codificador y el decodificador.

[0025] Un dispositivo de codificación de vídeo, tal como un codificador de vídeo o un decodificador de vídeo, puede usar las técnicas de la presente divulgación para realizar una combinación de predicción dependiente de la posición (PDPC), es decir, usar una o más ecuaciones parametrizadas que definen cómo combinar predicciones en base a valores de referencia filtrados y no filtrados, y en la posición del píxel predicho. La presente divulgación describe varios conjuntos de parámetros, de modo que el codificador puede probar cada uno de ellos (por ejemplo, por medio del uso de un análisis de tasa-distorsión) y señalar al decodificador los parámetros óptimos (por ejemplo, los parámetros que dan como resultado el mejor rendimiento de tasa-distorsión entre los parámetros que se prueban).

[0026] La FIG. 3 ilustra una predicción de un bloque 4x4 (p) usando una referencia no filtrada (r) de acuerdo con unas técnicas de la presente divulgación. La FIG. 4 ilustra una predicción de un bloque 4x4 (q) usando una referencia filtrada (s) de acuerdo con unas técnicas de la presente divulgación. Aunque las FIGS. 3 y 4 ilustran un bloque 4x4 de píxeles y 17 (4x4+1) valores de referencia respectivos, las técnicas de la presente divulgación se pueden aplicar a cualquier tamaño de bloque y número de valores de referencia.

[0027] Un codificador de vídeo que realiza una combinación de predicción dependiente de la posición puede utilizar una combinación entre las predicciones filtradas (q) y no filtradas (p), de modo que un bloque predicho para un bloque actual que se va a codificar se puede calcular usando valores de píxel tanto de las matrices de referencia filtradas (s) como de las no filtradas (r).

[0028] En un ejemplo de las técnicas de PDPC, dados dos conjuntos cualesquiera de predicciones de píxel $p_r[x, y]$ y $q_s[x, y]$, calculados usando solo las referencias no filtradas y filtradas r y s , respectivamente, el valor predicho combinado de un píxel, denotado por $v[x, y]$, se define por

$$v[x, y] = c[x, y] p_r[x, y] + (1 - c[x, y]) q_s[x, y] \quad (1)$$

donde $c[x, y]$ es el conjunto de parámetros de combinación. El valor del peso $c[x, y]$ puede ser un valor entre 0 y 1. La suma de los pesos $c[x, y]$ y $(1 - c[x, y])$ puede ser igual a uno.

[0029] En determinados ejemplos, puede no ser práctico tener un conjunto de parámetros tan grande como el número de píxeles del bloque. En dichos ejemplos, $c[x, y]$ se puede definir mediante un conjunto mucho más pequeño de parámetros, más una ecuación para calcular todos los valores de combinación a partir de esos parámetros. En dicho ejemplo, se puede usar un ejemplo de la siguiente fórmula:

$$v[x, y] = \left[\frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2 \left\lfloor \frac{y}{d_v} \right\rfloor} \right] + \left[\frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2 \left\lfloor \frac{x}{d_h} \right\rfloor} \right] + \left(\frac{N - \min(x, y)}{N} \right) g p_r^{(HEVC)}[x, y] + b[x, y] q_s^{(HEVC)}[x, y] \quad (2)$$

donde $c_1^v, c_2^v, c_1^h, c_2^h$, g y $d_v, d_h \in \{1, 2\}$ son parámetros de predicción, N es el tamaño de bloque, $p_r^{(HEVC)}[x, y]$ y $q_s^{(HEVC)}[x, y]$ son valores de predicción calculados usando la norma HEVC, para el modo específico, usando respectivamente las referencias no filtradas y filtradas, y

$$b[x, y] = 1 - \left[\frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right] - \left[\frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right] - \left(\frac{N - \min(x, y)}{N} \right) g \quad (3)$$

5 es un factor de normalización (es decir, para hacer que los pesos globales asignados a $p_r^{(HEVC)}[x, y]$ y $q_s^{(HEVC)}[x, y]$ sumen 1), definido por los parámetros de predicción.

[0030] La fórmula 2 se puede generalizar para cualquier norma de codificación de vídeo en la fórmula 2A:

$$v[x, y] = \left[\frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2^{\lfloor \frac{y}{d_v} \rfloor}} \right] + \left[\frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2^{\lfloor \frac{x}{d_h} \rfloor}} \right] + \left(\frac{N - \min(x, y)}{N} \right) g p_r^{(STD)}[x, y] + b[x, y] q_s^{(STD)}[x, y] \quad (2A)$$

10 donde $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v, d_h \in \{1, 2\}$, son parámetros de predicción, N es el tamaño de bloque, $p_r^{(STD)}[x, y]$ y $q_s^{(STD)}[x, y]$ son valores de predicción calculados usando una norma de codificación de vídeo (o sistema o algoritmo de codificación de vídeo), para el modo específico, usando respectivamente las referencias no filtradas y filtradas, y

$$15 \quad b[x, y] = 1 - \left[\frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right] - \left[\frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right] - \left(\frac{N - \min(x, y)}{N} \right) g \quad (3A)$$

es un factor de normalización (es decir, para hacer los pesos globales asignados a $p_r^{(STD)}[x, y]$ y $q_s^{(STD)}[x, y]$ sumen 1), definido por los parámetros de predicción.

20 **[0031]** Estos parámetros de predicción pueden incluir pesos para proporcionar una combinación lineal óptima de los términos predichos de acuerdo con el tipo de modo de predicción usado (por ejemplo, DC, plano y 33 modos direccionales de HEVC). Por ejemplo, la HEVC contiene 35 modos de predicción. Se puede construir una tabla de consulta con valores para cada uno de los parámetros de predicción $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v$ y d_h para cada uno de los modos de predicción (es decir, 35 valores de $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v$ y d_h para cada modo de predicción). Dichos valores se pueden codificar en un flujo de bits con el vídeo o pueden ser valores constantes conocidos de antemano por el codificador y el decodificador y que puede que no se necesiten transmitir en un archivo o flujo de bits. Los valores para $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v$ y d_h se pueden determinar mediante un algoritmo de entrenamiento de optimización encontrando los valores para los parámetros de predicción que proporcionan la mejor compresión para un conjunto de vídeos de entrenamiento. En otro ejemplo, hay una pluralidad de conjuntos de parámetros de predicción predefinidos para cada modo de predicción (por ejemplo, en una tabla de consulta) y el conjunto de parámetros de predicción seleccionado (pero no los parámetros en sí) se transmite a un decodificador en un archivo o flujo de bits codificados. En otro ejemplo, un codificador de vídeo puede generar sobre la marcha los valores para $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v$ y d_h y transmitirlos a un decodificador en un archivo o flujo de bits codificados.

35 **[0032]** En otro ejemplo, en lugar de usar una predicción HEVC, un dispositivo de codificación de vídeo que realiza estas técnicas puede usar una versión modificada de HEVC, como una que usa 65 predicciones direccionales en lugar de 33 predicciones direccionales. De hecho, se puede usar cualquier tipo de predicción intratrama.

40 **[0033]** En otro ejemplo, la fórmula se puede elegir para facilitar los cálculos. Por ejemplo, podemos usar el siguiente tipo de predictor

$$v[x, y] = \left[\frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2^{\lfloor y/d_v \rfloor}} \right] + \left[\frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2^{\lfloor x/d_h \rfloor}} \right] + b[x, y] p_{a,r,s}^{(HEVC)}[x, y] \quad (4)$$

donde

$$45 \quad b[x, y] = 1 - \left[\frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right] - \left[\frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right] \quad (5)$$

y

$$50 \quad p_{a,r,s}^{(HEVC)}[x, y] = a p_r^{(HEVC)}[x, y] + (1 - a) q_s^{(HEVC)}[x, y]. \quad (6)$$

[0034] Dicho enfoque puede aprovechar la linealidad de la predicción HEVC (u otra). Definiendo h como la respuesta

de impulso de un filtro k de un conjunto predefinido, si se tiene

$$\mathbf{s} = a \mathbf{r} + (1 - a)(\mathbf{h} * \mathbf{r}) \quad (7)$$

5 donde "*" representa convolución, entonces

$$p_{a,r,s}^{(\text{HEVC})}[x, y] = p_s^{(\text{HEVC})}[x, y] \quad (8)$$

es decir, la predicción combinada linealmente se puede calcular a partir de la referencia combinada linealmente.

10 **[0035]** Las fórmulas 4, 6 y 8 se pueden generalizar para cualquier norma de codificación de vídeo en las fórmulas 4A, 6A y 8A:

$$v[x, y] = \left[\frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2^{\lfloor y/d_v \rfloor}} \right] + \left[\frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2^{\lfloor x/d_h \rfloor}} \right] + b[x, y] p_{a,r,s}^{(\text{STD})}[x, y] \quad (4A)$$

15 donde

$$b[x, y] = 1 - \left[\frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right] - \left[\frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right] \quad (5A)$$

20 y

$$p_{a,r,s}^{(\text{STD})}[x, y] = a p_r^{(\text{STD})}[x, y] + (1 - a) q_s^{(\text{STD})}[x, y]. \quad (6A)$$

25 Dicho enfoque puede aprovechar la linealidad de la predicción de la norma de codificación. Definiendo \mathbf{h} como la respuesta de impulso de un filtro k de un conjunto predefinido, si se tiene

$$\mathbf{s} = a \mathbf{r} + (1 - a)(\mathbf{h} * \mathbf{r}) \quad (7A)$$

30 donde "*" representa convolución, entonces

$$p_{a,r,s}^{(\text{STD})}[x, y] = p_s^{(\text{STD})}[x, y] \quad (8A)$$

es decir, la predicción combinada linealmente se puede calcular a partir de la referencia combinada linealmente.

35 **[0036]** En un ejemplo, unas funciones de predicción pueden usar el vector de referencia (por ejemplo, \mathbf{r} y \mathbf{s}) solo como entrada. En este ejemplo, el comportamiento del vector de referencia no cambia, tanto si la referencia se ha filtrado como si no se ha filtrado. Si \mathbf{r} y \mathbf{s} son iguales (por ejemplo, determinada referencia \mathbf{r} no filtrada resulta ser la misma que otra referencia filtrada \mathbf{s}), entonces unas funciones predictivas, por ejemplo, $p[x, y]$ (también escrita como $p(x, y, \mathbf{r})$) es igual a $p_s[x, y]$ (también escrita como $p(x, y, \mathbf{s})$, aplicadas a unas referencias filtradas y no filtradas son iguales. Adicionalmente, las predicciones de píxel p y q pueden ser equivalentes (por ejemplo, generar la misma salida dada la misma entrada). En dicho ejemplo, las fórmulas (1)-(8) se pueden reescribir con la predicción de píxel $p[x, y]$ que reemplaza la predicción de píxel $q[x, y]$.

45 **[0037]** En otro ejemplo, la predicción (por ejemplo, los conjuntos de funciones) puede cambiar dependiendo de la información de que se ha filtrado una referencia. En este ejemplo, se pueden denotar diferentes conjuntos de funciones (por ejemplo, $p_r[x, y]$ y $q_s[x, y]$). En este caso, aunque \mathbf{r} y \mathbf{s} sean iguales, $p_r[x, y]$ y $q_s[x, y]$ pueden no ser iguales. En otras palabras, la misma entrada puede crear una salida diferente dependiendo de si la entrada se ha filtrado o no. En dicho ejemplo, tal vez no se pueda reemplazar $p[x, y]$ por $q[x, y]$.

50 **[0038]** Una ventaja de las ecuaciones de predicción mostradas es que, con la formulación parametrizada, se pueden determinar conjuntos de parámetros óptimos (es decir, los que optimizan la exactitud de la predicción), para diferentes tipos de texturas de vídeo, usando técnicas tales como el entrenamiento. Este enfoque, a su vez, se puede ampliar en algunos ejemplos calculando varios conjuntos de parámetros predictores para algunos tipos típicos de texturas, y teniendo un sistema de compresión donde el codificador prueba los predictores de cada conjunto, y codifica como información secundaria la que logra una mejor compresión.

60 **[0039]** La FIG. 5 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo 10 que puede utilizar unas técnicas descritas en esta divulgación, incluyendo PDPC. Como se muestra en la FIG. 5, el sistema de descodificación 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que un dispositivo de destino 14 va a descodificar en un momento posterior. En particular, el dispositivo de origen 12

proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualesquiera de entre una amplia gama de dispositivos, incluyendo ordenadores de escritorio, cuadernos digitales (es decir, ordenadores portátiles), ordenadores de tableta, descodificadores, equipos telefónicos manuales tales como los denominados teléfonos "inteligentes", los denominados teclados "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0040] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma 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 físicas de transmisión. 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 como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0041] En algunos ejemplos, los datos codificados se pueden facilitar desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, la interfaz de entrada puede acceder a los datos codificados desde el dispositivo de almacenamiento. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como un disco duro, unos discos Blu-ray, unos DVD, unos CD-ROM, una memoria flash, una memoria volátil o no volátil, u otros medios de almacenamiento digital cualesquiera adecuados para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados del dispositivo de almacenamiento por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor de FTP, unos dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de las mismas.

[0042] Las técnicas de esta divulgación no se limitan necesariamente a las aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua dinámica adaptativa a través de HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema de descodificación 10 puede estar configurado para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0043] En el ejemplo de la FIG. 5, el dispositivo de origen 12 incluye la fuente de vídeo 18, el codificador de vídeo 20 y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el descodificador de vídeo 30 y el dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas descritas en esta divulgación, tales como unas técnicas relacionadas con la intrapredicción de vídeo mejorada usando una combinación de predicción dependiente de la posición. De acuerdo con esta divulgación, el descodificador de vídeo 30 del dispositivo de destino 14 puede estar configurado para aplicar las técnicas descritas en esta divulgación, tales como unas técnicas relacionadas con la intrapredicción de vídeo mejorada usando una combinación de predicción dependiente de la posición. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde una fuente de vídeo externa 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede conectarse con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0044] Como se indica anteriormente, el dispositivo de origen 12 incluye una interfaz de salida 22 y el dispositivo de destino 14 incluye una interfaz de entrada 28. En algunos ejemplos, la interfaz de salida 22 representa un transmisor y la interfaz de entrada 28 representa un receptor. En otros ejemplos, la interfaz de salida 22 y la interfaz de entrada

28 representan ejemplos de transeptores (es decir, interfaces capaces de transmitir y recibir inalámbricamente señales de datos). Los transeptores pueden estar configurados para enviar y recibir datos de vídeo en señales inalámbricas. Por ejemplo, la interfaz de salida 22, cuando se implementa como un transeptor, puede enviar una señal de datos (por ejemplo, un medio legible por ordenador 16) que incluye datos de vídeo codificados, mientras que la interfaz de entrada 28, cuando se implementa como un transeptor, puede recibir una señal de datos (por ejemplo, un medio legible por ordenador 16) que incluye datos de vídeo codificados. Como se analiza anteriormente, el codificador de vídeo 20 puede proporcionar los datos de vídeo codificados a la interfaz de salida 22, mientras que la interfaz de entrada 28 puede proporcionar datos de vídeo codificados al descodificador de vídeo 30.

[0045] El sistema de descodificación ilustrado 10 de la FIG. 5 es simplemente un ejemplo. Las técnicas descritas en esta divulgación se pueden realizar mediante cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/descodificador de vídeo, típicamente denominado "CÓDEC". Además, las técnicas de esta divulgación también se pueden realizar mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación, en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y descodificación de vídeo. Por consiguiente, el sistema de descodificación 10 puede admitir la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

[0046] La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de transmisión de vídeo para recibir vídeo desde un proveedor de contenido de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. La interfaz de salida 22 puede facilitar a continuación la información de vídeo codificado a un medio legible por ordenador 16.

[0047] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una radiodifusión inalámbrica o una transmisión de red alámbrica, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificado desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificado al dispositivo de destino 14, por ejemplo, por medio de una transmisión de red. De forma similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampado de discos, puede recibir datos de vídeo codificado desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificado. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

[0048] La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde un medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, usada también por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, unos GOP. El dispositivo de visualización 32 visualiza los datos de vídeo descodificado ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0049] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, como las normas descritas anteriormente y, en algunos ejemplos, de acuerdo con la norma de codificación de vídeo de alta eficacia (HEVC), también denominada ITU-T H.265, o ampliaciones de la norma HEVC, o de acuerdo con la próxima generación de normas de codificación de vídeo. Las técnicas de esta divulgación, sin embargo, no están limitadas a ninguna norma de codificación en particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 5, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados cada uno con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si corresponde, las unidades MUX-DEMUX pueden adaptarse a un protocolo tal como el protocolo de multiplexor ITU H.223 u otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0050] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar implementados cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o combinaciones cualesquiera de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio legible por ordenador adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede estar incluido en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador combinado (CÓDEC) en un dispositivo respectivo.

[0051] En general, una trama o imagen de vídeo se puede dividir en una secuencia de bloques de árbol, que también se conocen como unidades de codificación más grandes (LCU), que pueden incluir muestras de luma y de croma. Los datos sintácticos dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad de codificación más grande en términos del número de píxeles. Un sector incluye un número de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo se puede dividir en uno o más sectores. Cada bloque de árbol se puede dividir en unidades de codificación (CU) de acuerdo con una estructura de árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, con un nodo raíz correspondiente al bloque de árbol. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

[0052] Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo se divide en sub-CU. Los elementos sintácticos para una CU se pueden definir de forma recursiva y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominan CU hoja incluso si no hay ninguna división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro sub-CU de tamaño 8x8 también se denominan CU hoja, aunque la CU de tamaño 16x16 no se haya dividido nunca.

[0053] Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto porque una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol se puede dividir en cuatro nodos hijo (también denominados sub-CU), y cada nodo hijo puede a su vez ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol se puede dividir, lo que se conoce como una profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación usa el término "bloque" para referirse a cualquiera de entre una CU, una unidad de predicción (PU) o una unidad de transformada (TU), en el contexto de la HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

[0054] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y en general tiene forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un tamaño máximo de 64x64 píxeles o mayor. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir dependiendo de cómo esté codificada la CU: en modo de omisión o directa, en modo de intrapredicción o en modo de interpretación. Las PU se pueden dividir para que no tengan forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

[0055] La norma HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. Las TU tienen típicamente un tamaño basado en el tamaño de las PU dentro de una CU determinada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT se pueden denominar unidades de transformada (TU). Los valores de diferencia de píxeles asociados a las TU se pueden transformar para generar coeficientes de transformada, que se pueden cuantificar.

[0056] Una CU hoja puede incluir una o más PU. En general, una PU representa un área espacial correspondiente a la totalidad o a una parte de la CU correspondiente, y puede incluir datos para recuperar y/o generar una muestra de referencia para la PU. Por otra parte, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en modo intra, los datos para la PU pueden estar incluidos en un árbol cuaternario residual (RQT), que puede incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. El RQT también se puede denominar árbol de transformada. En algunos ejemplos, el modo de intrapredicción se puede

señalar en la sintaxis de la CU hoja, en lugar del RQT. Como otro ejemplo, cuando la PU está codificada en modo inter, la PU puede incluir datos que definen información de movimiento, tales como uno o más vectores de movimiento, para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

[0057] Una CU hoja que tiene una o más PU también puede incluir una o más TU. Las unidades de transformada se pueden especificar usando un RQT (también denominado estructura de árbol cuaternario de TU), como se analiza anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformada. A continuación, cada unidad de transformada se puede dividir en otras sub-TU. Cuando una TU no se divide más, se puede denominar TU hoja. En general, para intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, el mismo modo de intrapredicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. Para intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no se limita necesariamente al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para intracodificación, una PU puede estar situada junto con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

[0058] Además, las TU de las CU hoja también pueden estar asociadas a unas respectivas estructuras de datos de árbol cuaternario, denominadas árboles cuaternarios residuales (RQT) o árboles de transformada, como se indica anteriormente. Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se divide la CU hoja en unas TU. El nodo raíz de un árbol cuaternario de TU en general corresponde a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU en general corresponde a un bloque de árbol (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta divulgación usa los términos CU y TU para referirse, respectivamente, a CU hoja y TU hoja, a menos que se indique lo contrario.

[0059] Una secuencia de vídeo típicamente incluye una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) en general comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir, en una cabecera del GOP, una cabecera de una o más de las imágenes o en otro lugar, datos sintácticos que describen un número de imágenes incluidas en el GOP. Cada sector de una imagen puede incluir datos sintácticos de sector que describen un modo de codificación para el respectivo sector. El codificador de vídeo 20 típicamente actúa en bloques de vídeo dentro de sectores de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

[0060] Como ejemplo, la predicción se puede realizar para PU de diversos tamaños. Suponiendo que el tamaño de una CU en particular es $2N \times 2N$, la intrapredicción se puede realizar en tamaños de PU de $2N \times 2N$ o $N \times N$, y la interpredicción se puede realizar en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. La división asimétrica para interpredicción también se puede realizar para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no se divide, mientras que la otra dirección se divide en un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % se indica con una "n" seguida de una indicación de "encima", "debajo", "izquierda" o "derecha". Por tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU $2N \times 2N$ que está dividida horizontalmente, con una PU $2N \times 0,5N$ en la parte superior y una PU $2N \times 1,5N$ en la parte inferior.

[0061] De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar configurados para intrapredicir un bloque de datos de vídeo usando una combinación lineal de un conjunto de valores de referencia filtrados y un conjunto de valores de referencia no filtrados, donde los valores de referencia corresponden a píxeles vecinos previamente decodificados. Es decir, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aplicar uno o más filtros a los píxeles vecinos para formar el conjunto de valores de referencia filtrados, y usar los propios píxeles vecinos como valores de referencia no filtrados. Además, la combinación lineal puede incluir aplicar conjuntos respectivos de pesos y/u otros parámetros de predicción a los valores de referencia filtrados y no filtrados. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden calcular un bloque predicho usando cualquiera de las fórmulas (1), (2) y (4) como se analiza anteriormente.

[0062] Más en particular, como se analiza en mayor detalle a continuación, el codificador de vídeo 20 en general predice y a continuación codifica datos de vídeo usando estas técnicas, y también puede determinar y a continuación señalar los parámetros de predicción que se van a usar durante la intrapredicción. El decodificador de vídeo 30, del mismo modo, recupera los parámetros de predicción del flujo de bits, cuando dichos parámetros de predicción están codificados en el flujo de bits, y a continuación aplica estas técnicas para predecir, decodificar y reconstruir datos de vídeo.

[0063] En esta divulgación, " $N \times N$ " y "N por N" se pueden usar de manera intercambiable para referirse a las dimensiones en píxeles de un bloque de vídeo, en términos de dimensiones verticales y horizontales, por ejemplo,

16x16 píxeles o 16 por 16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Del mismo modo, un bloque NxN en general tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no han de tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

[0064] Después de la codificación intrapredictiva o interpredictiva usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modo de generar datos de píxeles predictivos en el dominio espacial (también denominado dominio del píxel) y las TU pueden comprender coeficientes en el dominio de la transformada después de la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada entera, una transformada de *wavelet* o una transformada conceptualmente similar, a unos datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU para que incluyan coeficientes de transformada cuantificados representativos de los datos residuales para la CU. Es decir, el codificador de vídeo 20 puede calcular los datos residuales (en forma de un bloque residual), transformar el bloque residual para generar un bloque de coeficientes de transformada, y a continuación cuantificar los coeficientes de transformada para formar coeficientes de transformada cuantificados. El codificador de vídeo 20 puede formar una TU que incluye los coeficientes de transformada cuantificados, así como otra información sintáctica (por ejemplo, información de división para la TU).

[0065] Como se indica anteriormente, después de cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar una cuantificación de los coeficientes de transformada. La cuantificación en general se refiere a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. Por ejemplo, un valor de n bits se puede redondear a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

[0066] Después de la cuantificación, el codificador de vídeo 20 puede explorar los coeficientes de transformada, generando un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. La exploración se puede diseñar para que los coeficientes de mayor energía (y , por lo tanto, de menor frecuencia) se coloquen en la parte delantera de la matriz y para que los coeficientes de menor energía (y , por lo tanto, de mayor frecuencia) se coloquen en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para generar un vector en serie que se puede codificar entrópicamente. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar entrópicamente el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), la codificación entrópica de división de intervalos de probabilidad (PIPE) u otra metodología de codificación entrópica. El codificador de vídeo 20 también puede codificar entrópicamente elementos sintácticos asociados a los datos de vídeo codificados para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0067] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo que se va a transmitir. El contexto se puede referir, por ejemplo, a si los valores vecinos del símbolo son o no distintos de cero. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC se pueden construir de modo que unos códigos relativamente más cortos correspondan a símbolos más probables, mientras que unos códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede permitir un ahorro de bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de probabilidad puede estar basada en un contexto asignado al símbolo.

[0068] En general, el descodificador de vídeo 30 realiza un proceso sustancialmente similar, aunque recíproco, al realizado por el codificador de vídeo 20 para descodificar datos codificados. Por ejemplo, el descodificador de vídeo 30 realiza una cuantificación inversa y una transformada inversa de unos coeficientes de una TU recibida para reproducir un bloque residual. El descodificador de vídeo 30 usa un modo de predicción señalado (de intra- o interpredicción) para formar un bloque predicho. A continuación, el descodificador de vídeo 30 combina el bloque predicho y el bloque residual (de píxel en píxel) para reproducir el bloque original. Se puede realizar un procesamiento adicional, tal como realizar un proceso de eliminación de bloques para reducir artefactos visuales a lo largo de los límites de bloque. Además, el descodificador de vídeo 30 puede descodificar elementos sintácticos usando CABAC de una manera sustancialmente similar, aunque recíproca, al proceso de codificación CABAC del codificador de vídeo 20.

[0069] El codificador de vídeo 20 puede enviar además datos sintácticos, tales como datos sintácticos basados en

bloque, datos sintácticos basados en trama y datos sintácticos basados en GOP, al descodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de sector o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un número de tramas en el GOP respectivo, y los datos sintácticos de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

[0070] El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar, cada uno, como cualquiera de una variedad de circuitos codificadores o descodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno de entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede estar incluido en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye un codificador de vídeo 20 y/o un descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

[0071] La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas descritas en esta divulgación, tales como unas técnicas relacionadas con la intrapredicción de vídeo mejorada usando una combinación de predicción dependiente de la posición. De acuerdo con esta divulgación, el codificador de vídeo 20 puede estar configurado para aplicar las técnicas descritas en esta divulgación, tales como unas técnicas relacionadas con la intrapredicción de vídeo mejorada usando una combinación de predicción dependiente de la posición. El codificador de vídeo 20 puede realizar una intracodificación y una intercodificación de bloques de vídeo dentro de sectores de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en una predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) se puede referir a cualquiera de varios modos de codificación basados en el espacio. Los modos inter, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), se pueden referir a cualquiera de varios modos de codificación basados en el tiempo.

[0072] Como se muestra en la FIG. 6, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo que se va a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modo 40, una memoria de imágenes de referencia 64 (que también se puede denominar memoria intermedia de imágenes descodificadas (DPB)), un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación entrópica 56. La unidad de selección de modo 40, a su vez, incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de cuantificación inversa 58, una unidad de transformada inversa 60 y un sumador 62. El filtro de paso bajo 66 puede recibir información de imagen desde la memoria de imágenes de referencia 64 y puede filtrar previamente las muestras de referencia para su uso por la unidad de selección de modo 40 (y la unidad de intrapredicción 46). También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 6) para filtrar límites de bloques y eliminar las distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques podría filtrar típicamente la salida del sumador 62. También se pueden usar filtros adicionales (en el bucle o posteriores al bucle) además del filtro de eliminación de bloques y el filtro de paso bajo 66.

[0073] Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o sector de vídeo que se ha de codificar. La trama o el sector se pueden dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo recibido, en relación con uno o más bloques en una o más tramas de referencia, para proporcionar predicción temporal. La unidad de intrapredicción 46 puede realizar de forma alternativa la codificación intrapredictiva del bloque de vídeo recibido, en relación con uno o más bloques vecinos en la misma trama o sector que el bloque que se va a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuado para cada bloque de datos de vídeo.

[0074] El filtro de paso bajo 66 se puede aplicar a todos los bloques o, en algunos ejemplos, a bloques por encima de determinado tamaño (por ejemplo, bloques mayores de 4x4 en HEVC). En algunos ejemplos, el filtro de paso bajo 66 se puede aplicar solo a los píxeles de referencia. Se puede aplicar un filtro de paso bajo de 3 tomas a los bloques de datos de vídeo. Sin embargo, un experto en la técnica reconocerá que se puede usar cualquier número de tipos de filtros de paso bajo en base a las técnicas descritas.

[0075] En los ejemplos de la presente divulgación, se puede aplicar un filtro más fuerte a los valores de referencia de algunos o todos los bloques y combinarlos con los píxeles de referencia no filtrados y/o con filtrado de paso bajo para usar en una ecuación de predicción. A continuación, se analizan más detalles con respecto al uso de dichos filtros y la intrapredicción resultante realizada por la unidad de intrapredicción 46.

[0076] Además, la unidad de división 48 puede dividir bloques de datos de vídeo en subbloques, en base a la evaluación de sistemas de división previos en pasadas de codificación previas. Por ejemplo, la unidad de división 48 puede dividir inicialmente una trama o sector en unas LCU, y dividir cada una de las LCU en sub-CU en base al análisis

de tasa-distorsión (por ejemplo, optimización de tasa-distorsión). La unidad de selección de modo 40 puede generar además una estructura de datos de árbol cuaternario, indicativa de la división de una LCU en sub-CU. Las CU de nodo hoja del árbol cuaternario pueden incluir una o más PU y una o más TU.

5 **[0077]** La unidad de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, en base a resultados de errores, y proporciona el bloque predicho resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para su uso como trama de referencia. La unidad de selección de modo 40 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores de modo intra, información de división y otra información sintáctica de ese tipo, a la unidad de codificación entrópica 56.

15 **[0078]** La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que resulta coincidir estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, que se puede determinar mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel subentero de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionario de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y las posiciones de píxel fraccionario, y generar un vector de movimiento con una precisión de píxel fraccionario.

25 **[0079]** La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un sector intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar a partir 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.

35 **[0080]** La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar obtener o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, 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 vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando valores de píxel del bloque predictivo de los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles, como se analiza a continuación. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento en relación con unos componentes de luma, y la unidad de compensación de movimiento 44 usa unos vectores de movimiento calculados en base a los componentes de luma, tanto para componentes de croma como para componentes de luma. La unidad de selección de modo 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al sector de vídeo, para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del sector de vídeo.

40 **[0081]** La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción que se ha de usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar de entre los modos probados un modo de intrapredicción adecuado que se va a usar.

50 **[0082]** La unidad de intrapredicción 46 puede realizar la predicción a partir de un conjunto no filtrado o filtrado de píxeles de "referencia" vecinos, dependiendo del modo de predictor seleccionado y el tamaño de bloque usando una coordinación de predicción dependiente de la posición. Los píxeles de referencia filtrados se pueden filtrar por medio del filtro de paso bajo 66. La unidad de intrapredicción 46 puede usar una o más de un número de fórmulas ejemplares (1)-(8) para calcular un valor predicho combinado de un píxel.

60 **[0083]** En un ejemplo, la unidad de intrapredicción 46, cuando se dan dos conjuntos de predicciones de píxel $p_r[x, y]$ y $q_s[x, y]$, calculados usando solo las referencias no filtradas y filtradas r y s , respectivamente, puede calcular un valor predicho combinado de un píxel, denotado por $v[x, y]$, por medio de la fórmula (1) como se define anteriormente.

[0084] En otro ejemplo, la unidad de intrapredicción 46 puede utilizar una ecuación con un conjunto de parámetros más pequeño. En dichos ejemplos, $c[x, y]$ se puede definir mediante un conjunto mucho más pequeño de parámetros, más una ecuación para calcular todos los valores de combinación a partir de esos parámetros. En dicho ejemplo, la unidad de intrapredicción 46 puede usar la fórmula (2) como se define anteriormente.

[0085] En otro ejemplo, la fórmula se puede elegir para facilitar los cálculos. Por ejemplo, la unidad de intrapredicción 46 puede usar un predictor como se define en la fórmula (4) anterior. Dicho enfoque puede aprovechar la linealidad de la predicción HEVC (u otra) como se muestra en las fórmulas (7) y (8) anteriores.

[0086] La unidad de intrapredicción 46 puede seleccionar parámetros de predicción (por ejemplo, $c[x, y]$, c_1^v , c_2^v , c_1^h , c_2^h , g , dv , y o dh) que se corresponden con la ecuación de predicción usada por la unidad de intrapredicción 46 que mejor se ajusta a la predicción (por ejemplo, que tiene las mejores características de tasa-distorsión).

[0087] Por ejemplo, la unidad de selección de modo 40 puede calcular valores de tasa-distorsión usando un análisis de tasa-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción y los parámetros de predicción que tienen las mejores características de tasa-distorsión entre los modos probados. Usando el análisis de tasa-distorsión, la unidad de selección de modo 40 en general determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado, que se ha codificado para generar el bloque codificado, así como una tasa de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de selección de modo 40 puede calcular las proporciones a partir de las distorsiones y las tasas para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de tasa-distorsión para el bloque. Por ejemplo, la unidad de intrapredicción 46 puede probar cada una de un conjunto de ecuaciones parametrizadas combinando valores de referencia filtrados y no filtrados. La unidad de intrapredicción 46 puede probar cada uno (o una pluralidad) de los conjuntos de parámetros para determinar qué modo y parámetros presentan el mejor valor de tasa-distorsión para el bloque.

[0088] En un ejemplo, la unidad de selección de modo 40 puede utilizar una tabla que contiene parámetros de predicción para cada uno de los modos de predicción. La unidad de selección de modo 40 puede calcular los resultados de tasa-distorsión para cada procedimiento de combinación de predicción dependiente de la posición (PDPC), así como para no usar PDPC. La unidad de selección de modo 40 a continuación puede seleccionar el modo de predicción y los parámetros de predicción (o la ausencia de parámetros de predicción) que presenta el mejor valor de tasa-distorsión. En un ejemplo donde no se usa ninguna técnica de aceleración, el número de pruebas realizadas por la unidad de selección de modo 40 sería igual a (número de modos de predicción) x (número de conjuntos de parámetros PDPC + 1).

[0089] Después de seleccionar un modo de intrapredicción para un bloque, la unidad de selección de modo 40 puede proporcionar información indicativa del modo de intrapredicción seleccionado y los parámetros de predicción PDPC 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 vídeo 20 puede incluir en el flujo de bits transmitido datos de configuración, que pueden incluir una pluralidad de tablas de índices de modos de intrapredicción y una pluralidad de tablas modificadas de índices de modos de intrapredicción (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, parámetros de predicción (ya sean valores de parámetros de predicción o bien un único valor correspondiente a un conjunto predefinido de parámetros de predicción), una tabla de índices de modos de intrapredicción y una tabla modificada de índices de modos de intrapredicción que se va a usar para cada uno de los contextos. El codificador de vídeo 20 puede incluir parámetros de predicción en uno o más de un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), una cabecera de sector, una cabecera de bloque u otros de dichos datos.

[0090] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 40 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 52 aplica una transformada, tal como una transformada discreta de coseno (DCT) o una transformada conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores de coeficientes de transformada. En lugar de una DCT, se podrían usar transformadas de *wavelet*, transformadas enteras, transformadas de subbandas, transformadas discretas de seno (DST) u otros tipos de transformadas. En cualquier caso, la unidad de procesamiento de transformada 52 aplica la transformada al bloque residual, generando un bloque de coeficientes de transformada. La transformada puede convertir la información residual de un dominio del pixel a un dominio de la transformada, tal como un dominio de la frecuencia. La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir aún más la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación.

[0091] Después de la cuantificación, la unidad de codificación entrópica 56 explora y codifica entrópicamente los

coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación entrópica 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación entrópica de división en intervalos de probabilidad (PIPE), u otra técnica de codificación entrópica. En el caso de la codificación entrópica basada en contexto, el contexto puede estar basado en bloques vecinos. Después de la codificación entrópica mediante la unidad de codificación entrópica 56, el flujo de bits codificado se puede transmitir a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

[0092] La unidad de cuantificación inversa 58 y la unidad de transformada inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel. En particular, el sumador 62 suma el bloque residual reconstruido al bloque de predicción con compensación de movimiento, generado por la unidad de compensación de movimiento 44 o la unidad de intrapredicción 46, para generar un bloque de vídeo reconstruido para su 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 vídeo reconstruido como bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

[0093] El codificador de vídeo 20 en general usa el proceso analizado anteriormente para codificar cada bloque de cada imagen en una secuencia de vídeo codificada. Además, en algunos ejemplos, el codificador de vídeo 20 puede determinar las capas temporales a las cuales se va a asignar cada una de las imágenes. Además, el codificador de vídeo 20 puede estar configurado para codificar imágenes de otras capas, por ejemplo, otras vistas, capas de codificación de vídeo escalable, o similares. En cualquier caso, el codificador de vídeo 20 puede codificar además datos que indican una capa a la que pertenece cada imagen, para una o más capas (por ejemplo, de diversas dimensiones de vídeo).

[0094] La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar unas técnicas descritas en esta divulgación. De acuerdo con esta divulgación, el descodificador de vídeo 30 puede estar configurado para aplicar las técnicas descritas en esta divulgación, tales como unas técnicas relacionadas con la intrapredicción de vídeo mejorada usando una combinación de predicción dependiente de la posición. En el ejemplo de la FIG. 7, el descodificador de vídeo 30 incluye una unidad de descodificación entrópica 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformada inversa 78, una memoria de imágenes de referencia 82, un filtro de paso bajo 84 y un sumador 80. El descodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de descodificación en general recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 6). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación entrópica 70 (tanto directamente como por medio del filtro de paso bajo 84), mientras que la unidad de intrapredicción 74 puede generar datos de predicción en base a unos indicadores de modo de intrapredicción y unos parámetros de predicción recibidos desde la unidad de descodificación entrópica 70.

[0095] Durante el proceso de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un sector de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de descodificación entrópica 70 del descodificador de vídeo 30 descodifica entrópicamente el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción, parámetros de predicción PDPC y otros elementos sintácticos. La unidad de descodificación entrópica 70 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de sector de vídeo y/o el nivel de bloque de vídeo.

[0096] Cuando el sector de vídeo se codifica como un sector intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del sector de vídeo actual, en base a un modo de intrapredicción señalizado, unos datos de predicción y en unos datos de bloques descodificados previamente de la trama o imagen actual. Algunos o la totalidad de los bloques descodificados de la trama o imagen actual se filtran por medio del filtro de paso bajo 84 para su uso por la unidad de intrapredicción 74.

[0097] La unidad de intrapredicción 74 puede realizar la predicción a partir de un conjunto no filtrado o filtrado de píxeles de "referencia" vecinos, dependiendo del modo de predictor seleccionado y el tamaño de bloque usando una coordinación de predicción dependiente de la posición. Los píxeles de referencia filtrados se pueden filtrar por medio del filtro de paso bajo 84. La unidad de intrapredicción 74 puede usar una o más de un número de fórmulas ejemplares (1)-(8) definidas anteriormente para calcular un valor predicho combinado de un píxel.

[0098] En un ejemplo, la unidad de intrapredicción 74, cuando se dan dos conjuntos de predicciones de píxel $p_r[x, y]$ y $q_s[x, y]$, calculados usando solo las referencias no filtradas y filtradas \mathbf{r} y \mathbf{s} , respectivamente, puede calcular un valor predicho combinado de un píxel, denotado por $v[x, y]$, por medio de la fórmula (1) como se define anteriormente.

[0099] En otro ejemplo, la unidad de intrapredicción 74 puede utilizar una ecuación con un conjunto de parámetros más pequeño. En dichos ejemplos, $c[x, y]$ se puede definir mediante un conjunto mucho más pequeño de parámetros, más una ecuación para calcular todos los valores de combinación a partir de esos parámetros. En dicho ejemplo, la

unidad de intrapredicción 74 puede usar la fórmula (2) como se define anteriormente.

[0100] En otro ejemplo, la fórmula se puede elegir para facilitar los cálculos. Por ejemplo, la unidad de intrapredicción 74 puede usar un predictor como se define en la fórmula (4) anterior. Dicho enfoque puede aprovechar la linealidad de la predicción HEVC (u otra) como se muestra en las fórmulas (7) y (8) anteriores.

[0101] La unidad de intrapredicción 74 puede usar parámetros de predicción descodificados (por ejemplo, $c[x, y]$, c_1^v , c_2^v , c_1^h , c_2^h , g , d_v , y/o d_h) que corresponden a la ecuación de predicción usada por la unidad de intrapredicción 74 para calcular el bloque predicho.

[0102] Cuando la trama de vídeo se codifica como un sector intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del sector de vídeo actual, en base a los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de descodificación entrópica 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas en base a imágenes de referencia almacenadas en la memoria de imágenes de referencia 82. La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del sector de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está descodificando. 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 vídeo del sector de vídeo, un tipo de sector de interpredicción (por ejemplo, sector B, sector P o sector GPB), información de construcción para una o más de las listas de imágenes de referencia para el sector, vectores de movimiento para cada bloque de vídeo intercodificado del sector, el estado de interpredicción para cada bloque de vídeo intercodificado del sector y otra información para descodificar los bloques de vídeo en el sector de vídeo actual.

[0103] La unidad de compensación de movimiento 72 también puede realizar una interpolación en base a unos filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los que usa el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular unos valores interpolados para los píxeles subenteros de unos bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

[0104] La unidad de cuantificación inversa 76 realiza la cuantificación inversa, es decir, la descuantificación, de los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación entrópica 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_γ calculado por el descodificador de vídeo 30 para cada bloque de vídeo en el sector de vídeo, para determinar un grado de cuantificación y, de igual modo, un grado de cuantificación inversa que se debería aplicar.

[0105] La unidad de transformada inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de generar bloques residuales en el dominio del píxel.

[0106] Después de que la unidad de compensación de movimiento 72 o la unidad de intrapredicción 74 genere el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformada inversa 78 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72 o la unidad de intrapredicción 74. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados para eliminar las distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para allanar las transiciones de píxeles, o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dadas se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para una posterior compensación de movimiento. La memoria de imágenes de referencia 82 también almacena vídeo descodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 5.

[0107] El descodificador de vídeo 30 en general usa el proceso analizado anteriormente para descodificar cada bloque de cada imagen en una secuencia de vídeo codificada. Además, en algunos ejemplos, el descodificador de vídeo 30 puede descodificar datos que indican capas temporales a las que se asignan imágenes. Además, el descodificador de vídeo 30 puede estar configurado para descodificar imágenes de otras capas, por ejemplo, otras vistas, capas de codificación de vídeo escalable o similares. En cualquier caso, el descodificador de vídeo 30 además puede descodificar datos que indican una capa a la que pertenece cada imagen, para una o más capas (por ejemplo, de diversas dimensiones de vídeo).

[0108] La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo de acuerdo con las técnicas de esta divulgación. En un ejemplo de la divulgación, la unidad de intrapredicción 46 en el codificador de vídeo 20 puede estar configurada para codificar bloques de una imagen actual en un orden de barrido (100). El codificador de vídeo 20 puede estar configurado además para determinar la intrapredicción de un bloque actual de una imagen actual (102). El codificador de vídeo 20 puede estar configurado además para formar valores de referencia filtrados a partir de bloques descodificados vecinos del bloque actual (104). El filtro de paso bajo 66 puede estar configurado para filtrar los valores de referencia anteriores. El codificador de vídeo 20 puede estar configurado además para determinar parámetros de predicción para intraprededir el bloque actual (106). Determinar los parámetros de predicción puede incluir procesar diversos parámetros para determinar el conjunto de parámetros que es óptimo. Los ejemplos adicionales incluyen seleccionar un subconjunto de todos los parámetros de predicción y verificar cada combinación de parámetros del subconjunto para seleccionar un conjunto de parámetros que sea óptimo. La selección del conjunto de parámetros óptimo puede estar basado en un análisis de tasa-distorsión.

[0109] El codificador de vídeo 20 puede estar configurado además para intraprededir el bloque actual usando valores de referencia filtrados y valores de referencia no filtrados usando parámetros de predicción para formar el bloque predicho (108). En un ejemplo, el codificador de vídeo 20 puede estar configurado para generar un bloque predictivo calculando un conjunto de valores predictivos asociados con un conjunto de píxeles en base a una combinación lineal de uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada multiplicada por un primer conjunto de pesos y el uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada multiplicada por un segundo conjunto de pesos. En un ejemplo, calcular el conjunto de valores predictivos comprende calcular el conjunto de valores predictivos una pluralidad de veces con valores de parámetros diferentes y seleccionar un conjunto de parámetros óptimo de los valores de parámetros diferentes. El codificador de vídeo 20 puede estar configurado para generar el bloque predictivo por medio de cualquiera de las fórmulas paramétricas (1)-(8) analizadas anteriormente.

[0110] El codificador de vídeo 20 puede estar configurado además para calcular un bloque residual que representa las diferencias entre el bloque predicho y el bloque original (110). El codificador de vídeo 20 puede estar configurado además para transformar y cuantificar el bloque residual (112). El codificador de vídeo 20 puede estar configurado además para codificar entrópicamente coeficientes de transformada cuantificados, parámetros de predicción y una indicación del modo de intrapredicción para el bloque actual (114). El codificador de vídeo 20 puede estar configurado además para generar un flujo de bits con los coeficientes de transformada cuantificados codificados entrópicamente, los parámetros de predicción y la indicación de modo de intrapredicción para el bloque actual.

[0111] La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de ejemplo de acuerdo con las técnicas de esta divulgación. En un ejemplo de la divulgación, la unidad de intrapredicción 74 en el descodificador de vídeo 30 puede estar configurada para descodificar bloques de la imagen actual en un orden de barrido (200). El descodificador de vídeo 30 puede estar configurado además para descodificar entrópicamente coeficientes de transformada cuantificados, parámetros de predicción y una indicación de modo de intrapredicción para el bloque actual (202).

[0112] El descodificador de vídeo 30 puede estar configurado además para descuantificar el bloque residual (204).

[0113] El descodificador de vídeo 30 puede estar configurado además para formar valores de referencia filtrados a partir de bloques descodificados vecinos del bloque actual (206).

[0114] El descodificador de vídeo 30 puede estar configurado además para intraprededir el bloque actual usando valores de referencia filtrados y valores de referencia no filtrados usando parámetros de predicción descodificados para formar el bloque predicho (208). En un ejemplo, el descodificador de vídeo 30 puede estar configurado para calcular un conjunto de valores predictivos asociados con un conjunto de píxeles en base a una combinación lineal de uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada multiplicada por un primer conjunto de pesos y el uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada multiplicada por un segundo conjunto de pesos. En un ejemplo, el primer conjunto de pesos y el segundo conjunto de pesos varían en base a la posición de un píxel predicho del conjunto de píxeles. En otro ejemplo, un primer peso dentro del primer conjunto de pesos es mayor que un segundo peso del primer conjunto de pesos donde una primera distancia entre un primer valor predictivo del conjunto de valores predichos asociados con el primer peso y la matriz de referencia filtrada es mayor que una segunda distancia entre un segundo valor predictivo del conjunto de valores predichos asociados con el segundo peso y la matriz de referencia filtrada. En dicho ejemplo, en los píxeles predictivos dentro del bloque predictivo que están más lejos de los píxeles de referencia, el peso de los píxeles de referencia filtrados es mayor.

[0115] En otro ejemplo, el descodificador de vídeo 30 puede estar configurado para calcular un conjunto de valores de predicción no filtrados en base a la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada y calcular un conjunto de valores de predicción filtrados en base a la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada. El conjunto de valores de predicción no filtrados y el conjunto de valores de predicción filtrados pueden estar basado en uno o más parámetros de predictor direccionales (por ejemplo, modo de predicción direccional tal como los modos de predicción direccional HEVC descritos en la FIG. 2 anterior) descodificados a partir del flujo de bits.

[0116] En un ejemplo, el descodificador de vídeo 30 puede estar configurado para calcular el conjunto de valores predictivos por medio de fórmulas paramétricas (1)-(8) como se analiza anteriormente.

5 **[0117]** El descodificador de vídeo 30 puede estar configurado además para reconstruir un bloque de vídeo en base al conjunto calculado de valores predictivos (210).

10 **[0118]** Se ha de reconocer que dependiendo del ejemplo, determinados actos o actividades de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia diferente, se pueden añadir, fusionar u omitir por completo (por ejemplo, no todos los actos o las actividades descritas son necesarias para la puesta en práctica de las técnicas). Además, en determinados ejemplos, los actos o las actividades se pueden realizar de forma concurrente, por ejemplo, a través de subprocesamiento múltiple, procesamiento de interrupción o múltiples procesadores, en lugar de secuencialmente.

15 **[0119]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir a través de, un medio legible por ordenador como una o más instrucciones o código, y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de
20 almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador en general pueden corresponder a (1) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se puede acceder
25 mediante uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

30 **[0120]** A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otros dispositivos de almacenamiento en disco óptico, almacenamiento en disco magnético u otros tipos de almacenamiento magnético, memoria flash o cualquier otro medio que se puede usar para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión se denomina apropiadamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, un servidor o cualquier otro
35 origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debería entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que en su lugar están dirigidos a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen discos compactos (CD), discos láser, discos ópticos, discos
40 versátiles digitales (DVD), discos flexibles y discos Blu-ray, de los cuales los discos flexibles normalmente reproducen datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también deberían estar incluidos dentro del alcance de los medios legibles por ordenador.

45 **[0121]** Las instrucciones se pueden ejecutar mediante uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programable *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en
50 el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para codificar y descodificar, o incorporar en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

55 **[0122]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se
60 pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como se ha descrito anteriormente, junto con software y/o firmware adecuados.

65 **[0123]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento:

- 5 descodificar bloques vecinos de un bloque actual en una imagen de datos de vídeo;
- formar una predicción filtrada calculando una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados que comprenden versiones filtradas de píxeles vecinos del bloque actual en los bloques vecinos;
- 10 formar una predicción no filtrada calculando una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados que corresponden a versiones no filtradas de los píxeles vecinos;
- calcular un conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual; y
- 15 reconstruir el bloque actual en base al conjunto calculado de valores predictivos,

caracterizado por que el conjunto de valores predictivos se calcula en base a una combinación lineal de la predicción filtrada multiplicada por un primer conjunto de pesos y la predicción no filtrada multiplicada por un segundo conjunto de pesos.

2. El procedimiento de la reivindicación 1, en el que el primer conjunto de pesos y el segundo conjunto de pesos varían en base a una posición de un píxel predicho del conjunto de píxeles; y preferentemente en el que un primer peso dentro del primer conjunto de pesos es mayor que un segundo peso del primer conjunto de pesos cuando una primera distancia entre un primer valor predictivo del conjunto de valores predictivos asociados con el primer peso y la matriz de referencia filtrada es mayor que una segunda distancia entre un segundo valor predictivo del conjunto de valores predictivos asociados con el segundo peso y la matriz de referencia filtrada.

3. El procedimiento de la reivindicación 1, que comprende además:

- calcular un conjunto de valores predictivos no filtrados en base a la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada; y
- 35 calcular un conjunto de valores predictivos filtrados en base a la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada,
- en el que calcular el conjunto de valores predictivos comprende calcular el conjunto de valores predictivos usando el conjunto de valores predictivos no filtrados y el conjunto de valores predictivos filtrados; y preferentemente en el que
- 40 el conjunto de valores de predicción no filtrados y el conjunto de valores de predicción filtrados están basados en un modo de intracodificación direccional.

4. El procedimiento de la reivindicación 1, en el que calcular el conjunto de valores predichos comprende calcular el conjunto de valores predictivos por medio de una fórmula paramétrica:

$$v[x, y] = c[x, y] p_r[x, y] + (1 - c[x, y]) q_s[x, y],$$

50 en el que x representa un identificador de columna para un píxel del bloque actual e y representa un identificador de fila para el píxel del bloque actual, $v[x, y]$ representa un valor predictivo combinado para el píxel del bloque actual en una posición definida por x e y, $p_r[x, y]$ representa el uno o más de la pluralidad de valores de referencia no filtrados, $q_s[x, y]$ representa el uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada, c representa el segundo conjunto de pesos, y 1-c representa el primer conjunto de pesos.

55 5. El procedimiento de la reivindicación 1, en el que calcular el conjunto de valores predichos comprende calcular el conjunto de valores predictivos por medio de una fórmula paramétrica:

$$v[x, y] = \left[\frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2^{|y/d_v|}} \right] + \left[\frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2^{|x/d_h|}} \right] + \left(\frac{N - \text{mín}(x, y)}{N} \right) g p_r^{(STD)} [x, y] + b[x, y] q_s^{(STD)} [x, y]$$

60

donde x representa un identificador de columna para un píxel del bloque actual e y representa un identificador de fila para un píxel del bloque actual, $v[x, y]$ representa un valor predictivo combinado, r representa la matriz de referencia no filtrada, el segundo conjunto de pesos comprende $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v$ y d_h que son parámetros de predicción, N es un tamaño del bloque actual, $p_r^{(STD)}[x, y]$ representa la pluralidad de valores de referencia no filtrados calculados de acuerdo con una predicción no filtrada de una norma de codificación de vídeo, $q_s^{(STD)}[x, y]$ representa la pluralidad de valores de referencia filtrados calculados de acuerdo con una predicción filtrada de la norma de codificación de vídeo, el primer conjunto de pesos comprende $b[x, y]$ que representa un factor de normalización y $\min()$ representa una función mínimo.

5

10 6. El procedimiento de la reivindicación 4, en el que:

$$b[x, y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor - \left(\frac{N - \min(x, y)}{N} \right) g.$$

15

7. El procedimiento de la reivindicación 1, en el que calcular el conjunto de valores predichos comprende calcular el conjunto de valores predictivos por medio de una fórmula paramétrica:

$$v[x, y] = \left\lfloor \frac{c_1^{(v)} r[x, -1] - c_2^{(v)} r[-1, -1]}{2^{\lfloor y/d_v \rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1, y] - c_2^{(h)} r[-1, -1]}{2^{\lfloor x/d_h \rfloor}} \right\rfloor + b[x, y] p_{a,r,s}^{(STD)}[x, y],$$

donde

20

$$b[x, y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor,$$

y

25

$$p_{a,r,s}^{(STD)}[x, y] = a p_r^{(STD)}[x, y] + (1 - a) q_s^{(STD)}[x, y],$$

donde x representa un identificador de columna para un píxel del bloque actual e y representa un identificador de fila para un píxel del bloque actual, $v[x, y]$ representa un valor predictivo combinado, r representa la matriz de referencia no filtrada, el segundo conjunto de pesos comprende $a, c_1^v, c_2^v, c_1^h, c_2^h, d_v$ y d_h que son parámetros de predicción, el primer conjunto de pesos comprende $b[x, y]$, $p_r^{(STD)}[x, y]$ representa la pluralidad de valores de referencia no filtrados calculados de acuerdo con una predicción no filtrada de una norma de codificación de vídeo, $q_s^{(STD)}[x, y]$ representa la pluralidad de valores de referencia filtrados calculados de acuerdo con una predicción filtrada de la norma de codificación de vídeo.

30

35 8. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

recibir un bloque actual de una imagen, una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados, y una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados;

40

descodificar bloques vecinos de un bloque actual en una imagen de datos de vídeo;

formar una predicción filtrada calculando una matriz de referencia filtrada que comprende una pluralidad de valores de referencia filtrados que comprenden versiones filtradas de píxeles vecinos del bloque actual en los bloques vecinos;

45

formar una predicción no filtrada calculando una matriz de referencia no filtrada que comprende una pluralidad de valores de referencia no filtrados que corresponden a versiones no filtradas de los píxeles vecinos;

50

generar un bloque predictivo para el bloque actual, en el que generar comprende calcular un conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual;

generar un bloque residual en base a una diferencia entre el bloque actual y el bloque predictivo; y

55

codificar datos que representan el bloque residual en un flujo de bits;

caracterizado por que el conjunto de valores predictivos se calcula en base a una combinación lineal de la predicción filtrada multiplicada por un primer conjunto de pesos y la predicción no filtrada multiplicada por un segundo conjunto

de pesos.

9. El procedimiento de la reivindicación 8, en el que el cálculo del conjunto de valores predictivos comprende:

- 5 calcular una pluralidad de conjuntos de valores predictivos usando diferentes valores de parámetros; y
 seleccionar uno de los conjuntos de valores predictivos como un conjunto óptimo de valores predictivos.

10. El procedimiento de la reivindicación 9, en el que seleccionar el uno de los conjuntos de valores predictivos como el conjunto óptimo de valores predictivos comprende:

- realizar un análisis de tasa-distorsión en la pluralidad de conjuntos de valores predictivos; y
15 seleccionar el uno de los conjuntos de valores predictivos que logra el mejor rendimiento de tasa-distorsión entre la pluralidad de conjuntos de valores predictivos.

11. El procedimiento de la reivindicación 10, en el que generar un bloque predictivo para el bloque actual comprende además:

- 20 calcular un segundo conjunto de valores predictivos asociados con un conjunto de píxeles del bloque actual en base a uno o más de la pluralidad de valores de referencia filtrados en la matriz de referencia filtrada o el uno o más de la pluralidad de valores de referencia no filtrados en la matriz de referencia no filtrada;
 realizar un segundo análisis de tasa-distorsión en el segundo conjunto de valores predictivos; y
25 seleccionar uno del segundo conjunto de valores predictivos y el conjunto óptimo de valores predictivos.

12. El procedimiento de la reivindicación 9, que comprende además codificar los valores de parámetros correspondientes al uno seleccionado de los conjuntos de valores predictivos en el flujo de bits.

13. Un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo:

- una memoria configurada para almacenar datos de vídeo asociados con una imagen; y
35 uno o más procesadores en comunicación con la memoria y configurados para llevar a cabo las etapas de procedimiento de las reivindicaciones 1 a 7.

14. Un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo:

- 40 una memoria configurada para almacenar vídeo asociado con una imagen; y uno o más procesadores en comunicación con la memoria y configurados para llevar a cabo las etapas de procedimiento de las reivindicaciones 8 a 12.

15. Un medio no transitorio legible por ordenador que tiene almacenadas instrucciones para procesar datos de vídeo que, cuando se ejecutan, hacen que uno o más procesadores lleven a cabo las etapas de cualquiera de las reivindicaciones de procedimiento 1 a 12.

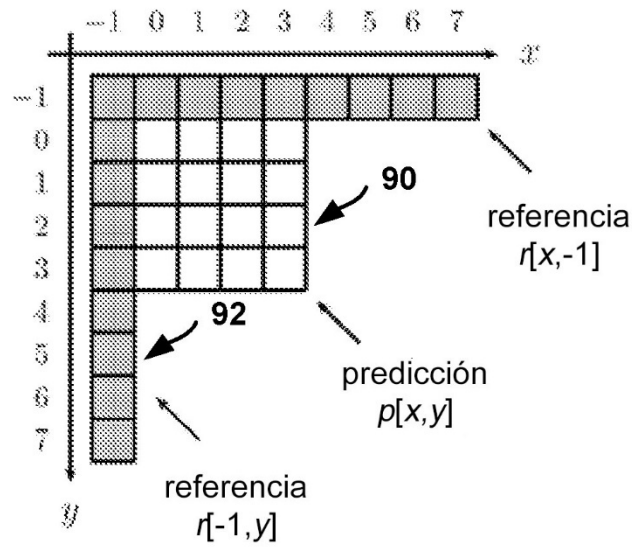


FIG. 1

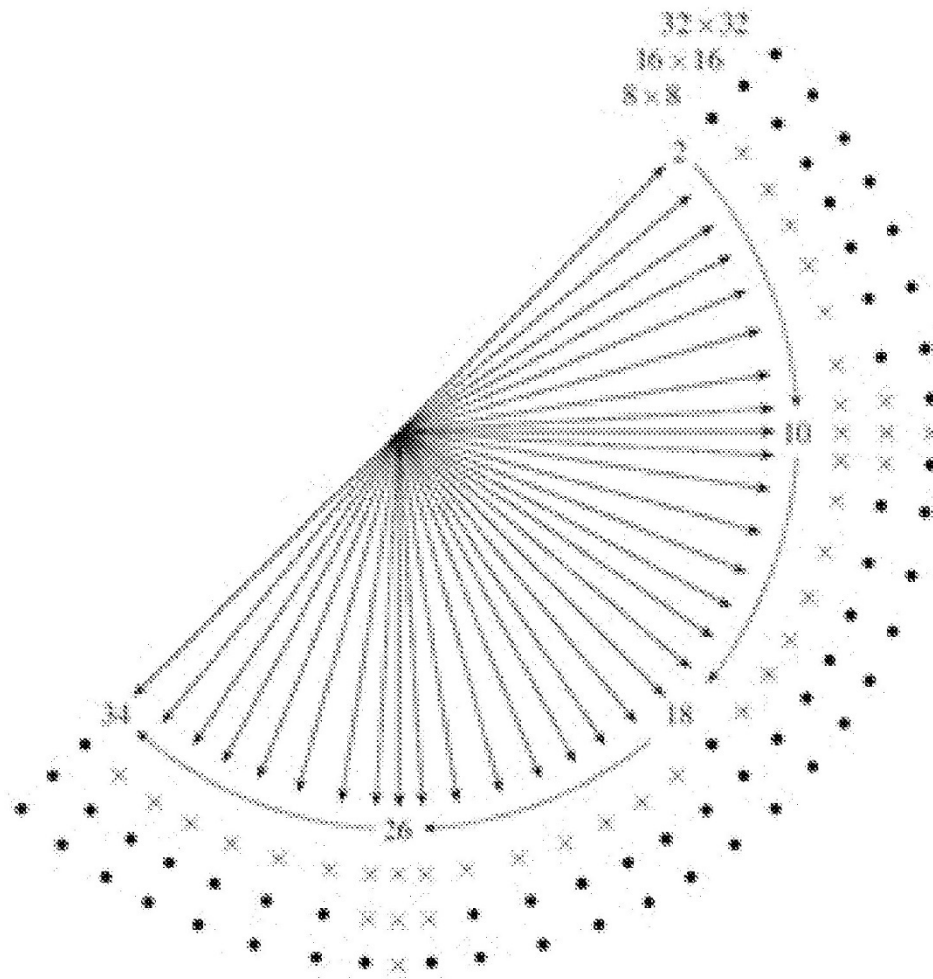


FIG. 2

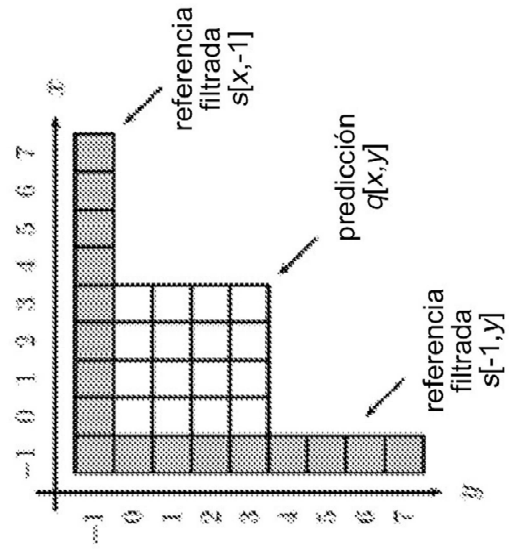


FIG. 4

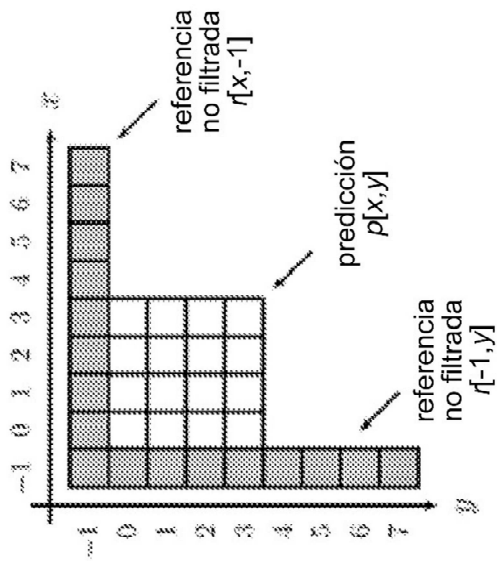


FIG. 3

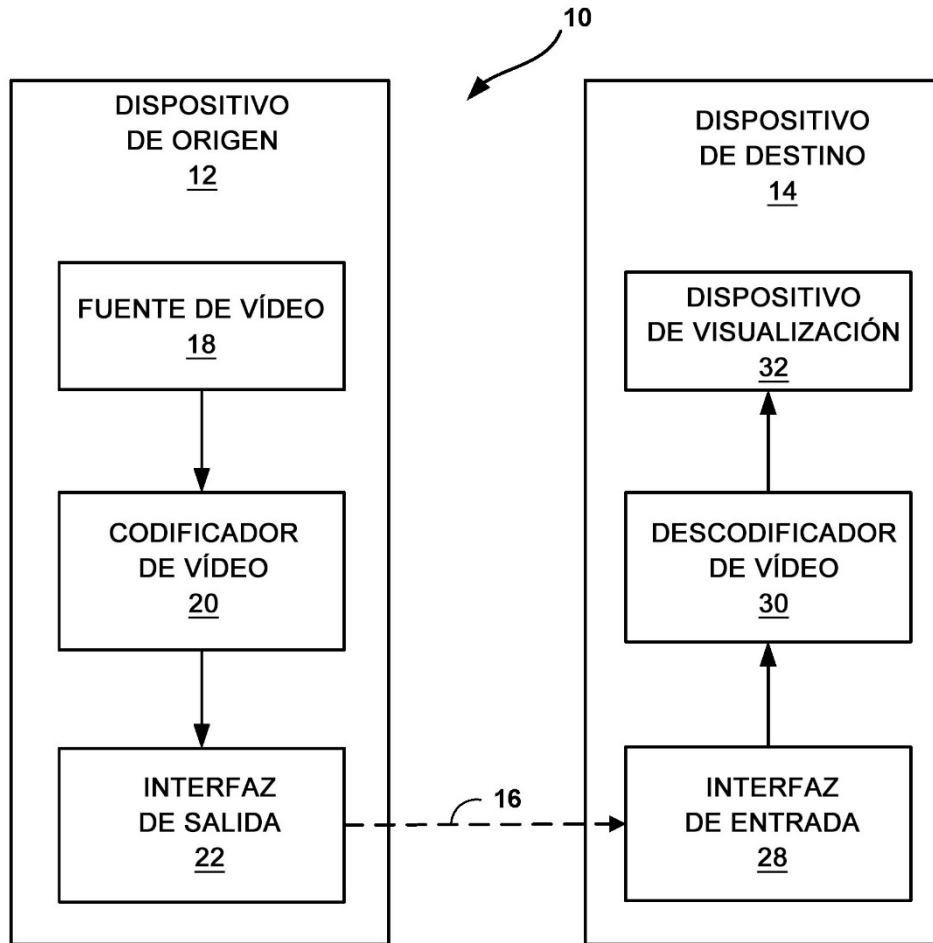


FIG. 5

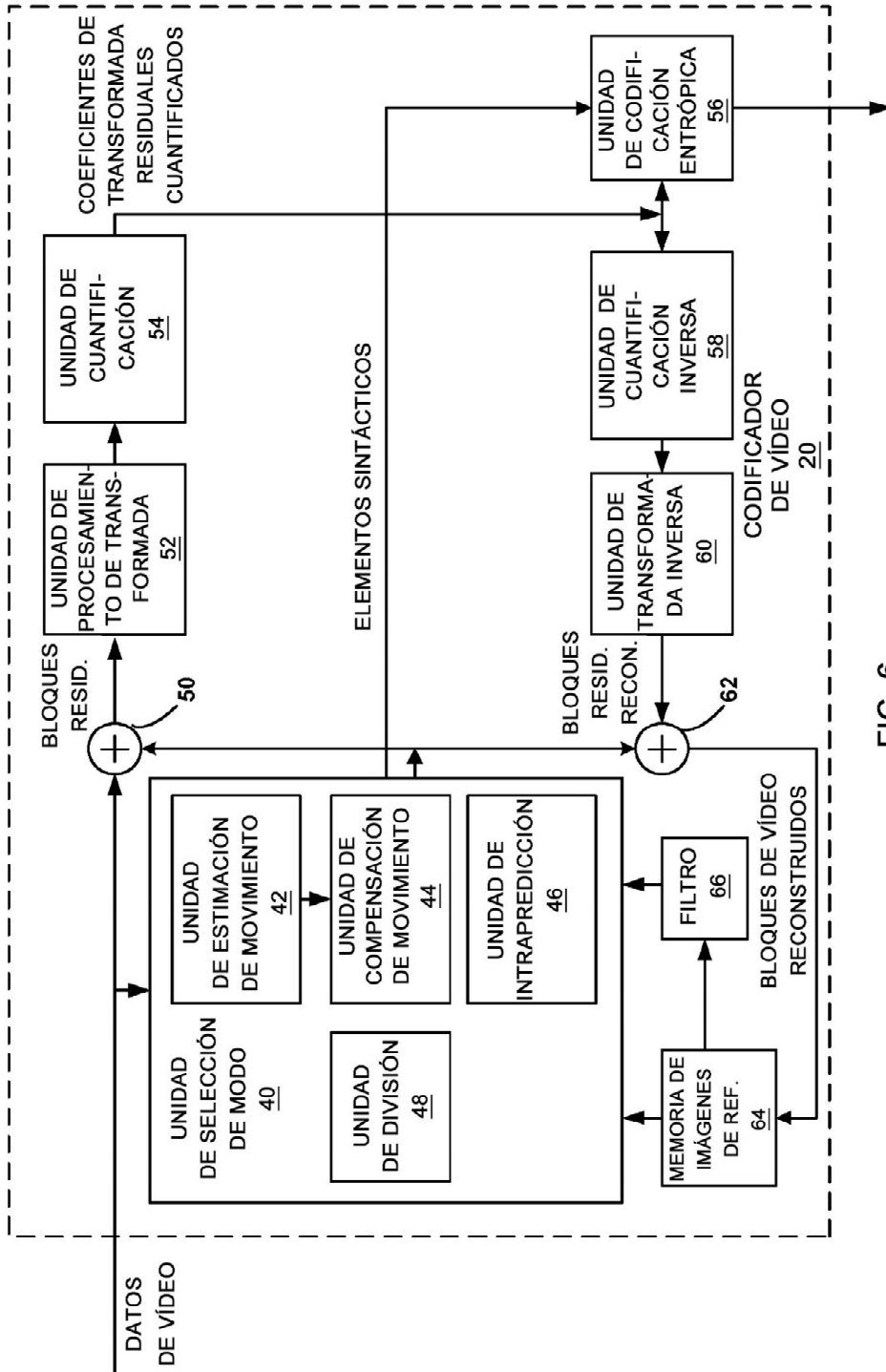


FIG. 6

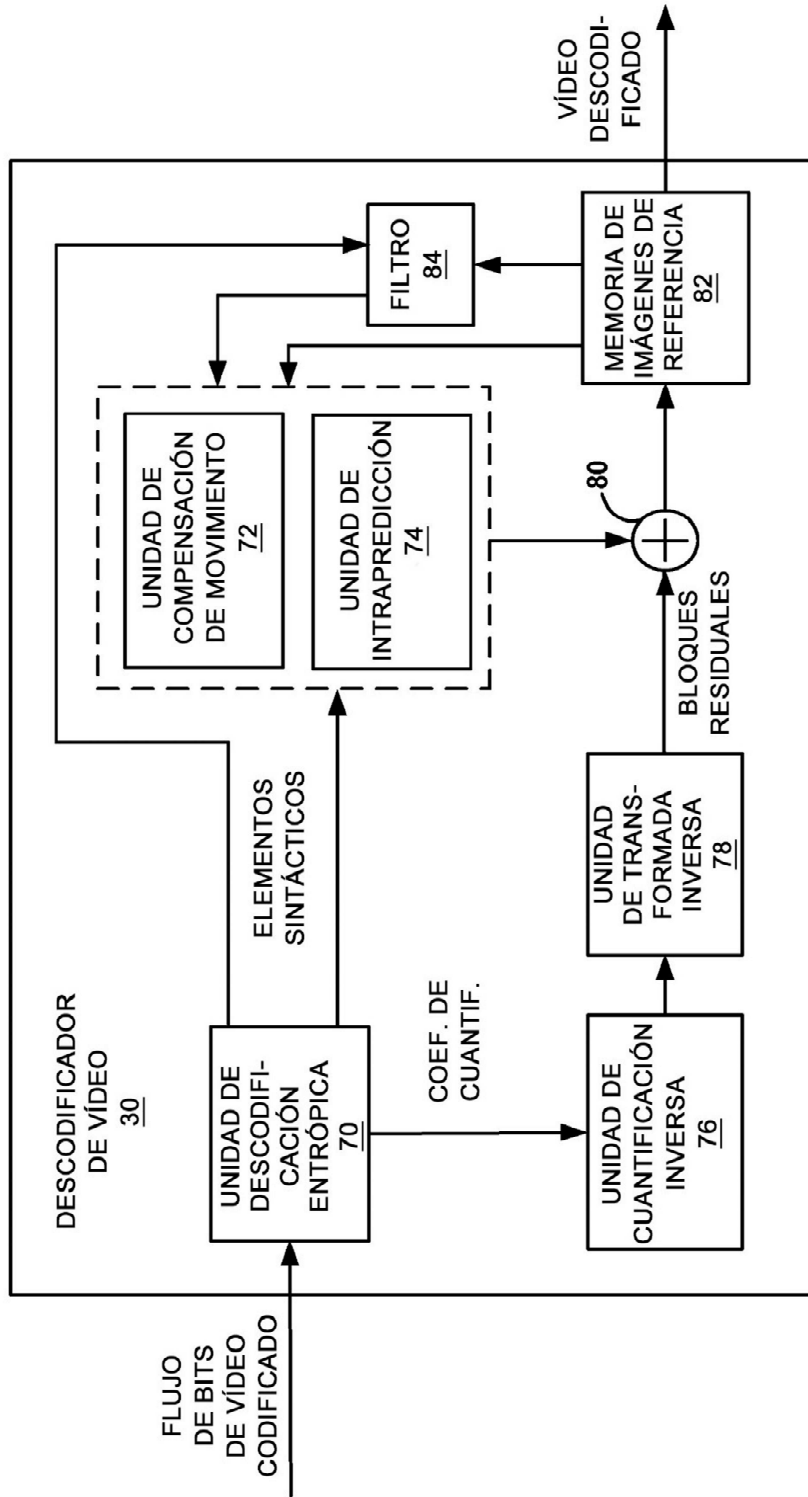


FIG. 7

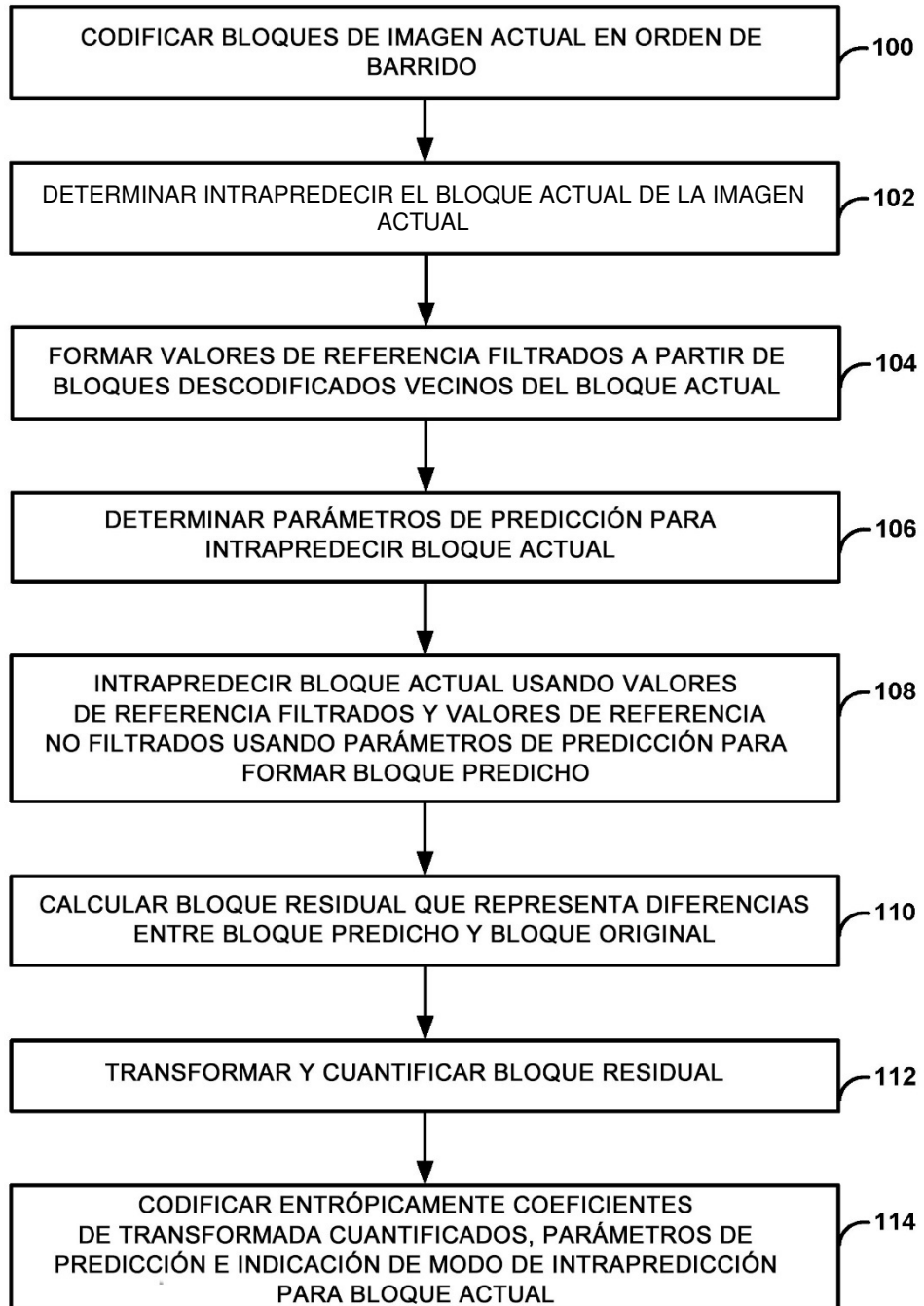


FIG. 8

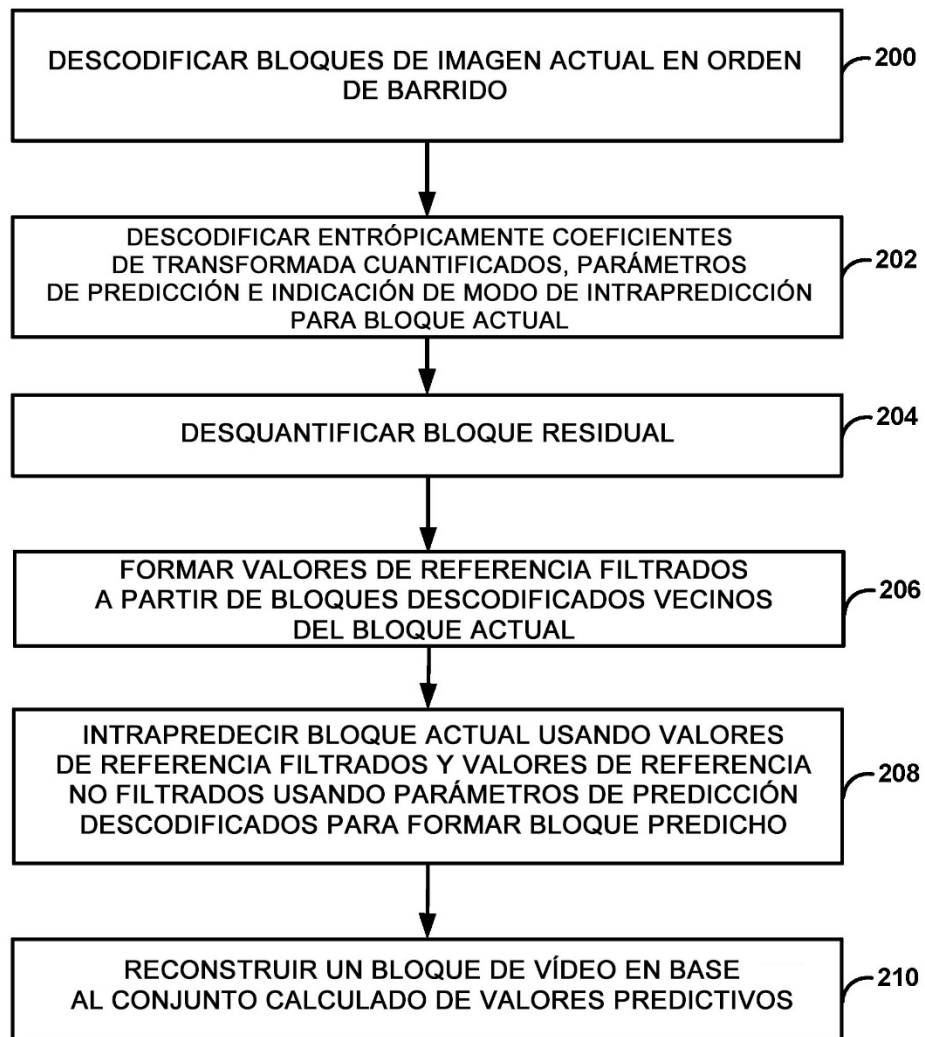


FIG. 9