

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 649 532**

51 Int. Cl.:

H04N 19/597 (2014.01)

H04N 19/105 (2014.01)

H04N 19/70 (2014.01)

H04N 19/136 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **02.10.2014 PCT/US2014/058822**

87 Fecha y número de publicación internacional: **09.04.2015 WO15051118**

96 Fecha de presentación y número de la solicitud europea: **02.10.2014 E 14793352 (7)**

97 Fecha y número de publicación de la concesión europea: **13.09.2017 EP 3053340**

54 Título: **Predicción ponderada explícita de alta precisión para la codificación de vídeo**

30 Prioridad:

03.10.2013 US 201361886230 P

17.10.2013 US 201361892198 P

01.10.2014 US 201414504097

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

12.01.2018

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)
International IP Administration, 5775 Morehouse
Drive**

San Diego, California 92121-1714, US

72 Inventor/es:

**PU, WEI;
CHEN, JIANLE;
KARCZEWICZ, MARTA;
KIM, WOO-SHIK y
SOLE ROJALS, JOEL**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 649 532 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

Predicción ponderada explícita de alta precisión para la codificación de vídeo

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

10 ANTECEDENTES

10 Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores
 digitales, sistemas de difusión directa digital, sistemas de difusión 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
 15 videojuegos, teléfonos móviles o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de
 videoconferencia, dispositivos de transmisión por flujo de vídeo y similares. Los dispositivos de vídeo digital
 implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2,
 MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, Codificación de Vídeo Avanzada (AVC), la norma de
 20 Codificación de Vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo y las 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.

25 Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (entre
 imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de
 vídeo basada en bloques, un fragmento de vídeo (por ejemplo, un fotograma de vídeo o una parte de un fotograma
 de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades de
 codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un fragmento intracodificado (I) de una imagen
 se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos de la misma
 30 imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción
 espacial con respecto a muestras de referencia en bloques vecinos de la misma imagen o predicción temporal con
 respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse fotogramas,
 y las imágenes de referencia pueden denominarse fotogramas de referencia.

35 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que va a codificarse. Los
 datos residuales representan diferencias de píxeles entre el bloque original que va a codificarse 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
 40 intracodificación y los datos residuales. Para compresión adicional, los datos residuales pueden transformarse desde
 el dominio de píxeles a un dominio de transformada, dando como resultado coeficientes de transformada residuales,
 que posteriormente pueden cuantificarse. Los coeficientes de transformada cuantificados, dispuestos inicialmente en
 una matriz bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes de
 transformada, y puede aplicarse codificación entrópica para lograr aún más compresión.

45 RESUMEN

Por lo general, esta divulgación describe técnicas para la predicción ponderada explícita de alta precisión. Estas
 técnicas pueden usarse para una ampliación de rango para codificación de vídeo de alta eficiencia (HEVC), o
 50 ampliaciones a otras normas de codificación de vídeo.

En un ejemplo, un procedimiento de descodificación de datos de vídeo incluye descodificar un valor para un
 elemento de sintaxis representativo de si se habilita una profundidad de bits alta para datos de vídeo, cuando el
 valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta: descodificar un valor para un
 elemento de sintaxis representativo de la profundidad de bits alta para uno o más parámetros de los datos de vídeo,
 55 descodificar valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de
 bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits alta, y descodificar los
 datos de vídeo basado al menos en parte en los valores para los parámetros.

En otro ejemplo, un procedimiento de codificación de datos de vídeo incluye determinar la habilitación de una
 60 profundidad de bits alta para datos de vídeo, después de determinar la habilitación de la profundidad de bits alta:
 codificar un valor para un elemento de sintaxis que indica que está habilitada la profundidad de bits alta, codificar un
 valor para un elemento de sintaxis representativo de la profundidad de bits alta para uno o más parámetros de los
 datos de vídeo, codificar valores para los parámetros de tal modo que los valores para los parámetros tengan
 profundidades de bits basadas en la profundidad de bits alta, y codificar los datos de vídeo basado al menos en
 65 parte en los valores para los parámetros.

- En otro ejemplo, un dispositivo para codificar datos de vídeo incluye una memoria configurada para almacenar datos de vídeo, y un codificador de vídeo configurado para codificar un valor para un elemento de sintaxis representativo de si se habilita una profundidad de bits alta para los datos de vídeo, y cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta: codifica un valor para un elemento de sintaxis representativo de la profundidad de bits alta para uno o más parámetros de los datos de vídeo, codifica valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits alta, y codifica los datos de vídeo basado al menos en parte en los valores para los parámetros.
- En otro ejemplo, un dispositivo para codificar datos de vídeo incluye medios para codificar un valor para un elemento de sintaxis representativo de si se habilita una profundidad de bits alta para datos de vídeo, medios para codificar un valor para un elemento de sintaxis representativo de la profundidad de bits alta para uno o más parámetros de los datos de vídeo cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta, medios para codificar valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits alta cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta, y medios para codificar los datos de vídeo basado al menos en parte en los valores para los parámetros cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta.
- En otro ejemplo, un medio de almacenamiento legible por ordenador (por ejemplo, un medio de almacenamiento legible por ordenador no transitorio) ha almacenado en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador codifique un valor para un elemento de sintaxis representativo de si está habilitada una profundidad de bits alta para datos de vídeo, cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta: codifica un valor para un elemento de sintaxis representativo de la profundidad de bits alta para uno o más parámetros de los datos de vídeo, codifica valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits alta, y codifica los datos de vídeo basado al menos en parte en los valores para los parámetros.
- Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción a continuación. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

- La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar técnicas para predicción ponderada explícita de alta precisión.
- La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para predicción ponderada explícita de alta precisión.
- La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas para predicción ponderada explícita de alta precisión.
- La FIG. 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar un bloque actual de acuerdo con las técnicas de esta divulgación.
- La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para descodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

- Por lo general, las técnicas de esta divulgación se refieren a la predicción ponderada explícita de alta precisión. Por ejemplo, estas técnicas pueden usarse en una ampliación de rango de codificación de vídeo de alta eficiencia (HEVC). HEVC se describe en ITUT H.265, SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIA, Infraestructura de Servicios Audiovisuales-Codificación de Vídeo en Movimiento, "Codificación de Vídeo de Alta Eficiencia", abril de 2013 (en lo sucesivo en el presente documento, "H.265"). Se han propuesto diversas ampliaciones a HEVC. Una de dichas ampliaciones es la ampliación de rango de HEVC, descrita en "especificación de texto de Ampliaciones de Rango de Codificación de Vídeo de Alta Eficiencia (HEVC): Borrador 4", JCTVC-N1005_v3, agosto de 2013 (en lo sucesivo en el presente documento, "JCTVC-N1005").

- H.265 especifica una versión de perfil principal I, que describe códecs de 8 bits y 10 bits. Los parámetros relacionados con la predicción ponderada explícita en H.265 están restringidos al códec de 8 bits. Sin embargo, como se propone para la ampliación de rango de HEVC, el vídeo de entrada puede ser de hasta 16 bits, lo que hace que los parámetros de predicción ponderada explícita de 8 bits sean insuficientes. Esta divulgación describe técnicas de predicción ponderada explícita mejoradas que pueden usarse para hacer coincidir la profundidad de bits de

parámetros de predicción ponderada explícita con la profundidad de bits del vídeo de entrada.

La sección 7.3.6.3 de JCTVC-N1005 especifica sintaxis para parámetros de predicción ponderada como se muestra en la tabla 1 a continuación. Los parámetros de predicción ponderada, que en la tabla 1 forman parte de una tabla de ponderación de predicción (pred_weight_table), pueden señalarse en un encabezado de fragmento.

5

TABLA 1

	Descriptor
pred_weight_table() {	
luma_log2_weight_denom	ue(v)
if(ChromaArrayType != 0)	
delta_chroma_log2_weight_denom	se(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++)	
luma_weight_l0_flag[i]	u(1)
pred_weight_table(){	
if(ChromaArrayType != 0)	
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++)	
chroma_weight_l0_flag[i]	u(1)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {	
if(luma_weight_l0_flag[i]) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if(chroma_weight_l0_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
if(slice_type == B) {	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++)	
luma_weight_l1_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++)	
chroma_weight_l1_flag[i]	u(1)
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_weight_l1_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	
}	

JCTVC-N1005, en la sección 7.4.7.3, define la semántica para los elementos de sintaxis de la tabla 1 como sigue:

luma_log2_weight_denom es el logaritmo en base 2 del denominador para todos los factores de ponderación de luma. El valor de luma_log2_weight_denom estará en el intervalo de 0 a 7, inclusive.

5 delta_chroma_log2_weight_denom es la diferencia del logaritmo en base 2 del denominador para todos los factores de ponderación de croma.

10 La variable ChromaLog2WeightDenom se obtiene para que sea igual a luma_log2_weight_denom + delta_chroma_log2_weight_denom, y el valor estará en el intervalo de 0 a 7, inclusive.

luma_weight_l0_flag[i] igual a 1 especifica que los factores de ponderación para el componente luma de predicción de lista 0 que usa RefPicList0[i] están presentes. luma_weight_l0_flag[i] igual a 0 especifica que estos factores de ponderación no están presentes.

15 chroma_weight_l0_flag[i] igual a 1 especifica que los factores de ponderación para los valores de predicción de croma de predicción de lista 0 que usan RefPicList0[i] están presentes. chroma_weight_l0_flag[i] igual a 0 especifica que estos factores de ponderación no están presentes. Cuando no está presente chroma_weight_l0_flag[i], se deduce que es igual a 0.

20 delta_luma_weight_l0[i] es la diferencia del factor de ponderación aplicado al valor de predicción de luma para la predicción de lista 0 que usa RefPicList0[i].

25 La variable LumaWeightL0[i] se obtiene para que sea igual a $(1 \ll \text{luma_log2_weight_denom}) + \text{delta_luma_weight_l0}[i]$. Cuando luma_weight_l0_flag[i] es igual a 1, el valor de delta_luma_weight_l0[i] estará en el intervalo de -128 a 127, inclusive. Cuando luma_weight_l0_flag[i] es igual a 0, se deduce que LumaWeightL0[i] es igual a $2^{\text{luma_log2_weight_denom}}$.

30 luma_offset_l0[i] es la compensación aditiva aplicada al valor de predicción de luma para la predicción de lista 0 que usa RefPicList0[i]. El valor de luma_offset_l0[i] estará en el intervalo de -128 a 127, inclusive. Cuando luma_weight_l0_flag[i] es igual a 0, se deduce luma_offset_l0[i] igual a 0.

35 delta_chroma_weight_l0[i][j] es la diferencia del factor de ponderación aplicado a los valores de predicción de croma para la predicción de lista 0 que usa RefPicList0[i] con j igual a 0 para Cb y j igual a 1 para Cr.

40 La variable ChromaWeightL0[i][j] se obtiene para que sea igual a $(1 \ll \text{ChromaLog2WeightDenom}) + \text{delta_chroma_weight_l0}[i][j]$. Cuando chroma_weight_l0_flag[i] es igual a 1, el valor de delta_chroma_weight_l0[i][j] estará en el intervalo de -128 a 127, inclusive. Cuando chroma_weight_l0_flag[i] es igual a 0, se deduce que ChromaWeightL0[i][j] es igual a $2^{\text{ChromaLog2WeightDenom}}$.

delta_chroma_offset_l0[i][j] es la diferencia de la compensación aditiva aplicada a los valores de predicción de croma para la predicción de lista 0 que usa RefPicList0[i] con j igual a 0 para Cb y j igual a 1 para Cr.

La variable ChromaOffsetL0[i][j] se obtiene como sigue:

45

$$\text{ChromaOffsetL0}[i][j] = \text{Clip3}(-128, 127, (\text{delta_chroma_offset_l0}[i][j] - ((128 * \text{ChromaWeightL0}[i][j]) \gg \text{ChromaLog2WeightDenom}) + 128)) \quad (7-50)$$

50 El valor de delta_chroma_offset_l0[i][j] estará en el intervalo de -512 a 511, inclusive. Cuando chroma_weight_l0_flag[i] es igual a 0, se deduce que ChromaOffsetL0[i][j] es igual a 0.

55 luma_weight_l1_flag[i], chroma_weight_l1_flag[i], delta_luma_weight_l1[i], luma_offset_l1[i], delta_chroma_weight_l1[i][j] y delta_chroma_offset_l1[i][j] tienen la misma semántica que luma_weight_l0_flag[i], chroma_weight_l0_flag[i], delta_luma_weight_l0[i], luma_offset_l0[i], delta_chroma_weight_l0[i][j] y delta_chroma_offset_l0[i][j], respectivamente, con 10, L0, lista 0 y List0 reemplazados por 11, L1, lista 1 y List1, respectivamente.

60 La variable sumWeightLOFlags se obtiene para que sea igual a la suma de luma_weight_l0_flag[i] + 2 * chroma_weight_l0_flag[i], para i = 0..num_ref_idx_l0_active_minus1.

Cuando slice_type es igual a B, la variable sumWeightL1Flags se obtiene para que sea igual a la suma de luma_weight_l1_flag[i] + 2 * chroma_weight_l1_flag[i], para i = 0..num_ref_idx_l1_active_minus1.

65 JCTVC-N1005 requiere, para conformidad con el flujo de bits, que cuando slice_type es igual a P, sumWeightLOFlags será menor o igual a 24, y cuando slice_type es igual a B, la suma de sumWeightLOFlags y

sumWeightL1Flags será menor o igual a 24.

La sección 8.5.3.3.4.3 de JCTVC-N1005 especifica el proceso de predicción de muestra ponderada explícita como:

5 Las entradas para este proceso son:

- dos variables nPbW y nPbH que especifican la anchura y la altura del bloque de predicción de luma,
- dos matrices (nPbW)x(nPbH) predSamplesLO y predSamplesL1,
- los indicadores de utilización de lista de predicción, predFlagLO y predFlagL1,
- 10 ▪ los índices de referencia, refldxL0 y refldxL1,
- una variable cldx que especifica el índice de componente de color,
- una profundidad de bits de muestras, bitDepth.

15 La salida de este proceso es la matriz (nPbW)x(nPbH) predSamples de los valores de muestra de predicción.

La variable shift1 se establece igual a Max(2, 14 - bitDepth).

Las variables log2Wd, o0, o1 y w0, w1 se obtienen como sigue:

- 20 ▪ Si cldx es igual a 0 para muestras de luma, se aplica lo siguiente:

$$10g2Wd = luma_log2_weight_denom + shift1 \quad (8-243)$$

$$w0 = LumaWeightL0[refldxL0] \quad (8-244)$$

$$25 \quad w1 = lumaWeightL1[refldxL1] \quad (8-245)$$

$$o0 = luma_offset_l0[refldxL0] * (1 \ll (bitDepth - 8)) \quad (8-246)$$

$$30 \quad o1 = luma_offset_l1[refldxL1] * (1 \ll (bitDepth - 8)) \quad (8-247)$$

- De lo contrario (cldx no es igual a 0 para muestras de croma), se aplica lo siguiente:

$$35 \quad log2Wd = ChromaLog2WeightDenom + shift1 \quad (8-248)$$

$$w0 = ChromaWeightL0[refldxL0][cldx - 1] \quad (8-249)$$

$$w1 = ChromaWeightL1[refldxL1][cldx - 1] \quad (8-250)$$

$$40 \quad o0 = ChromaOffsetL0[refldxL0][cldx - 1] * (1 \ll (bitDepth - 8)) \quad (8-251)$$

$$o1 = ChromaOffsetL1[refldxL1][cldx - 1] * (1 \ll (bitDepth - 8)) \quad (8-252)$$

La muestra de predicción predSamples[x][y] con $x = 0..nPbW - 1$ e $y = 0..nPbH - 1$ se obtienen como sigue:

- 45 ▪ Si el predFlagLO es igual a 1 y predFlagL1 es igual a 0, los valores de muestra de predicción se obtienen como sigue:

$$50 \quad \begin{aligned} & \text{if}(log2Wd \geq 1) \\ & \quad \text{predSamples}[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, \\ & \quad \quad ((\text{predSamplesLO}[x][y] * w0 + 2^{\log2Wd} - 1) \gg \log2Wd) + o0) \end{aligned} \quad (8-253)$$

else

$$55 \quad \text{predSamples}[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, \text{predSamplesLO}[x][y] * w0 + o0)$$

- -De lo contrario, si el predFlagLO es igual a 0 y predFlagL1 es igual a 1, los valores de muestra de predicción se obtienen como sigue:

$$60 \quad \begin{aligned} & \text{if}(log2Wd \geq 1) \\ & \quad \text{predSamples}[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, \\ & \quad \quad ((\text{predSamplesL1}[x][y] * w1 + 2^{\log2Wd} - 1) \gg \log2Wd) + o1) \end{aligned} \quad (8-254)$$

else

$$65 \quad \text{predSamples}[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, \text{predSamplesL1}[x][y] * w1 + o1)$$

- De lo contrario (predFlagLO es igual a 1 y predFlagL1 es igual a 1), los valores de muestra de predicción se obtienen como sigue:

```

5      predSamples[ x ][ y ] = Clip3( 0, ( 1 << bitDepth ) - 1,
      ( predSamplesL0 [ x ][ y ] * w0 + predSamplesL1[ x ][ y ] * w1 +
      ( ( o0 + o1 + 1 ) << log2Wd ) ) >> ( log2Wd + 1 ) )      (8-255)

```

Esta divulgación describe técnicas que pueden mejorar la predicción ponderada explícita de JCTVC-N1005. Por ejemplo, las técnicas de esta divulgación pueden implementarse para la predicción ponderada explícita de alta precisión. La predicción ponderada explícita de alta precisión puede mejorar la eficiencia de codificación para vídeo de entrada de profundidad de bits alta. De acuerdo con estas técnicas, un codificador de vídeo (por ejemplo, un codificador de vídeo o un descodificador de vídeo) puede aplicar de forma adaptativa un desplazamiento a la derecha de acuerdo con la profundidad de bits de datos de entrada para garantizar que no haya desbordamiento usando multiplicación de 16 bits. Además, las técnicas de esta divulgación pueden incluir señalización de un valor para un elemento de sintaxis (tal como un indicador) indicativo de si se usa la predicción ponderada explícita de alta precisión.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo 10 de ejemplo que puede utilizar técnicas para la predicción ponderada explícita de alta precisión. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de vídeo codificados que se van a descodificar posteriormente mediante un dispositivo de destino 14. En particular, el dispositivo fuente 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, cajas de conexión, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", tabletas, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión por flujo de vídeo o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden estar equipados para comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar a través del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación que habilita al dispositivo fuente 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o alámbrico, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal 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 fuente 12 al dispositivo de destino 14.

En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 a un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios de almacenamiento digital adecuados cualesquiera para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento a través de flujo o descarga. El servidor de ficheros 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. Servidores de ficheros de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados por medio de cualquier conexión de datos estándar, incluyendo una conexión a internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de wifi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión por flujo, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo de internet, tales como el flujo adaptativo dinámico sobre

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 10 puede configurarse para admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión por flujo de vídeo, reproducción de vídeo, difusión de vídeo y/o videotelefonía.

5 En el ejemplo de la FIG. 1, el dispositivo fuente 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 fuente 12 puede estar configurado para aplicar las técnicas para la predicción ponderada explícita de alta precisión. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de vídeo desde una fuente de vídeo externa 18 tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

15 El sistema ilustrado 10 de la FIG. 1 es simplemente un ejemplo. Las técnicas para la predicción ponderada explícita de alta precisión pueden realizarse 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 pueden realizarse mediante un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también pueden realizarse mediante un preprocesador de vídeo. El dispositivo fuente 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo fuente 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de tal modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y descodificación de vídeo. Así que, el sistema 10 puede admitir transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión por flujo de vídeo, reproducción de vídeo, difusión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo captado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo fuente, 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 videocámara, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o alámbricas. En cada caso, el vídeo captado, precaptado o generado por ordenador puede codificarse mediante el codificador de vídeo 20. La información de vídeo codificada puede emitirse posteriormente mediante la interfaz de salida 22 en un medio legible por ordenador 16.

40 El medio legible por ordenador 16 puede incluir medios temporales, tales como una difusión inalámbrica o una transmisión de red alámbrica, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, a través de transmisión por red. De forma similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de estampación de discos, puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

50 La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información de sintaxis definida por el codificador de vídeo 20, que también usa el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, GOP. El dispositivo de visualización 32 muestra los datos de vídeo descodificado a 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 emisores de luz orgánicos (OLED) u otro tipo de dispositivo de visualización.

60 El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse al modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, Codificación de Vídeo Avanzada (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263.

Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados cada uno con un codificador y un descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos independientes. Si es el caso, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

La norma ITU-T H.264/MPEG-4 (AVC) la formuló el Grupo de Expertos de Codificación de Vídeo de ITU-T (VCEG) junto al Grupo de Expertos de Imágenes en Movimiento de ISO/IEC (MPEG) como el producto de una asociación colectiva conocida como el Equipo de Vídeo Conjunto (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la recomendación de ITU-T H.264, Codificación de Vídeo Avanzada para servicios audiovisuales genéricos, por el Grupo de Estudio de ITU-T, con fecha de marzo de 2005, que se puede denominar en el presente documento la norma H.264 o especificación H.264, o la norma o especificación H.264/AVC. El Equipo de Vídeo Conjunto (JVT) continúa trabajando en ampliaciones para H.264/MPEG-4 AVC.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitería de codificador adecuada, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), lógica discreta, software, hardware, firmware o cualesquiera combinaciones de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 puede incluirse en uno o más codificadores o descodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

El JCT-VC continúa desarrollando la norma HEVC. Las actividades de normalización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción.

Por lo general, el modelo de funcionamiento del HM describe que un fotograma o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de mayor tamaño (LCU) (también denominadas "unidades arboladas de codificación") que incluyen muestras tanto de luma como de croma. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad de codificación de mayor tamaño en lo que respecta al número de píxeles. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Un fotograma o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede separarse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por lo general, una estructura de datos de árbol cuaternario incluye un nodo por CU, donde un nodo raíz corresponde al bloque arbolado. Si una CU se separa 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.

Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de separación, que indica si la CU correspondiente al nodo está separada en sub-CU. Los elementos de sintaxis para una CU pueden definirse de manera recursiva, y pueden depender de si la CU está separada en sub-CU. Si una CU no se separa adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hoja incluso si no hay separación explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se separa adicionalmente, las cuatro sub-CU 8x8 también se denominarán CU hoja aunque la CU 16x16 no se haya separado nunca.

Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede separarse en cuatro nodos secundarios (también denominados sub-CU) y cada nodo secundario puede a su vez ser un nodo principal y separarse en otros cuatro nodos secundarios. Un nodo secundario final, no separado, denominado un nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado una CU hoja. Los datos de sintaxis asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede separarse un bloque arbolado, denominado profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad de codificación de menor tamaño (SCU). Esta divulgación usa el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de 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).

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma

cuadrada. El tamaño de la CU puede oscilar desde 8x8 píxeles hasta el tamaño del bloque arbolado con un máximo de 64x64 píxeles o mayor. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo de salto o directo, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU pueden dividirse para no tener forma cuadrada. Los datos de sintaxis 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).

La norma HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU tienen típicamente el mismo tamaño o uno más pequeño que las PU. En algunos ejemplos, muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformada, que pueden cuantificarse.

Una CU hoja puede incluir una o más unidades de predicción (PU). Por lo 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 una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en intramodo, pueden incluirse datos para la PU en un árbol cuaternario residual (RQT), que pueden incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en intermodo, la PU puede incluir datos que definen 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 píxeles de un cuarto o precisión de píxeles de un octavo), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, la lista 0, la lista 1 o la lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden especificarse usando un RQT (también denominado estructura de árbol cuaternario de TU), como se analizó anteriormente. Por ejemplo, un indicador de separación puede indicar si una CU hoja está separada en cuatro unidades de transformación. Posteriormente, cada unidad de transformación puede separarse adicionalmente en sub-TU adicionales. Cuando una TU no se separa adicionalmente, puede denominarse 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 está necesariamente limitada al tamaño de una PU. De este modo, las TU pueden ser más grandes o más pequeñas que una PU. Para intracodificación, una PU puede estar coubicada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder con el tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse a las estructuras de datos de árbol cuaternario respectivas, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo está dividida la CU hoja en varias TU. El nodo raíz de un árbol cuaternario de TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU corresponde en general a un bloque arbolado (o LCU). Las TU del RQT que no están separadas se denominan TU hoja. Por lo general, esta divulgación usa los términos CU y TU para referirse a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

Una secuencia de vídeo incluye típicamente una serie de fotogramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende en general una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos de sintaxis en un encabezado del GOP, en un encabezado de una o más de las imágenes, o en otras ubicaciones, que describen un número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos de sintaxis de fragmento que describen un modo de codificación para el fragmento respectivo. Un codificador de vídeo funciona típicamente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de 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.

Como ejemplo, el HM admite predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el HM admite intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$ e interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para interpredicción en 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 está dividida, mientras que la otra dirección está dividida en un 25 % y un 75 %. La parte de la CU correspondiente a la división de un 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo",

“izquierda” o “derecha”. De este modo, por ejemplo, “ $2N \times N$ U” 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.

5 En esta divulgación, “ $N \times N$ ” y “ N por N ” pueden usarse indistintamente para referirse a las dimensiones de píxeles de un bloque de vídeo en lo que respecta a dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. Por lo general, un bloque 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque $N \times N$ tiene en general 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, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

15 Tras 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 de sintaxis que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado el dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformada, tras la aplicación de una transformada, por ejemplo, una transformada de coseno discreta (DCT), una transformada de números enteros, una transformada de ondículas o una transformada conceptualmente similar a los 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 incluyendo los datos residuales para la CU, y posteriormente transformar las TU para producir coeficientes de transformada para la CU.

20 Tras cualquier transformación para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. Cuantificación se refiere en general a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede redondearse hacia abajo hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

25 Tras la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformada, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. La exploración puede diseñarse para colocar coeficientes de energía más alta (y por lo tanto de frecuencia más baja) en la parte frontal de la matriz y para colocar coeficientes de energía más baja (y por lo tanto de frecuencia más alta) 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 producir un vector en serie que pueda codificarse de manera entrópica. 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 de manera entrópica el vector unidimensional, por ejemplo, codificación de longitud variable adaptativa de acuerdo con el contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), codificación entrópica de división de intervalos de probabilidad (PIPE) u otra metodología de codificación entrópica. El codificador de vídeo 20 también puede codificar de manera entrópica elementos de sintaxis asociados a los datos de vídeo codificados para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

30 Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo contextual a un símbolo que va a transmitirse. El contexto puede referirse, por ejemplo, a si los valores vecinos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que va a transmitirse. Las palabras de código en la VLC pueden construirse de tal modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que va a transmitirse. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

35 De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar la predicción ponderada explícita de alta precisión. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar datos de vídeo de acuerdo con cierta semántica modificada a los elementos de sintaxis de la tabla 1 analizados anteriormente. Como ejemplo, la semántica modificada puede incluir cualquiera o toda la semántica mostrada a continuación. En la semántica para los elementos de sintaxis a continuación, el texto subrayado representa diferencias respecto a la semántica expuesta en JCTVC-N1005.

40 $\text{luma_log2_weight_denom}$ es el logaritmo en base 2 del denominador para todos los factores de ponderación de luma. El valor de $\text{luma_log2_weight_denom}$ estará en el intervalo de 0 a $\text{bitDepth}-1$, inclusive.

45 $\text{delta_chroma_log2_weight_denom}$ es la diferencia del logaritmo en base 2 del denominador para todos los factores

de ponderación de croma.

La variable ChromaLog2WeightDenom se obtiene para que sea igual a luma_log2_weight_denom + delta_chroma_log2_weight_denom, y el valor estará en el intervalo de 0 a bitDepth-1, inclusive.

5 delta_luma_weight_10[i] es la diferencia del factor de ponderación aplicado al valor de predicción luma para la predicción de lista 0 que usa RefPicList0[i].

10 La variable LumaWeightL0[i] se obtiene para que sea igual a (1 << luma_log2_weight_denom) + delta_luma_weight_10[i]. Cuando luma_weight_10_flag[i] es igual a 1, el valor de delta_luma_weight_10[i] estará en el intervalo de $\frac{-2^{\text{bitDepth}-1}}{2^{\text{luma_log2_weight_denom}}}$ a $2^{\text{bitDepth}-1}-1$, inclusive. Cuando luma_weight_10_flag[i] es igual a 0, se deduce que LumaWeightL0[i] es igual a $2^{\text{luma_log2_weight_denom}}$.

15 luma_offset_10[i] es la compensación aditiva aplicada al valor de predicción de luma para la predicción de lista 0 que usa RefPicList0[i]. El valor de luma_offset_10[i] estará en el intervalo de $-2^{\text{bitDepth}-1}$ a $2^{\text{bitDepth}-1}-1$, inclusive. Cuando luma_weight_10_flag[i] es igual a 0, se deduce luma_offset_10[i] igual a 0.

20 delta_chroma_weight_10[i][j] es la diferencia del factor de ponderación aplicado a los valores de predicción de croma para la predicción de lista 0 que usa RefPicList0[i] con j igual a 0 para Cb y j igual a 1 para Cr.

25 La variable ChromaWeightL0[i][j] se obtiene para que sea igual a (1 << ChromaLog2WeightDenom) + delta_chroma_weight_10[i][j]. Cuando chroma_weight_10_flag[i] es igual a 1, el valor de delta_chroma_weight_10[i][j] estará en el intervalo de $\frac{-2^{\text{bitDepth}-1}}{2^{\text{ChromaLog2WeightDenom}}}$ a $2^{\text{bitDepth}-1}-1$, inclusive. Cuando chroma_weight_10_flag[i] es igual a 0, se deduce que ChromaWeightL0[i][j] es igual a $2^{\text{ChromaLog2WeightDenom}}$.

delta_chroma_offset_10[i][j] es la diferencia de la compensación aditiva aplicada a los valores de predicción de croma para la predicción de lista 0 que usa RefPicList0[i] con j igual a 0 para Cb y j igual a 1 para Cr.

30 La variable ChromaOffsetL0[i][j] se obtiene como sigue:

$$\text{ChromaOffsetL0}[i][j] = \text{Clip3}\left(\frac{-2^{\text{bitDepth}-1}}{2^{\text{ChromaLog2WeightDenom}}}, \frac{2^{\text{bitDepth}-1}-1}{2^{\text{ChromaLog2WeightDenom}}}, \left(\text{delta_chroma_offset_10}[i][j] - \left(\frac{2^{\text{bitDepth}-1}}{2^{\text{ChromaLog2WeightDenom}}}\right) * \text{ChromaWeightL0}[i][j]\right)\right) \quad (7-50)$$

35 El valor de delta_chroma_offset_10[i][j] estará en el intervalo de $\frac{2^{\text{bitDepth}+1}}{2^{\text{ChromaLog2WeightDenom}}}$ a $2^{\text{bitDepth}+1}-1$, inclusive. Cuando chroma_weight_10_flag[i] es igual a 0, se deduce que ChromaOffsetL0[i][j] es igual a 0.

40 delta_luma_weight_11[i], luma_offset_11[i], delta_chroma_weight_11[i][j] y delta_chroma_offset_11[i][j] tienen la misma semántica que delta_luma_weight_10[i], luma_offset_10[i], delta_chroma_weight_10[i][j] y delta_chroma_offset_10[i][j], respectivamente, con l0, L0, lista 0 y List0 reemplazados por l1, L1, lista 1 y List1, respectivamente.

45 Además o como alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para implementar un proceso de predicción de muestra ponderada explícita de alta precisión como se describe a continuación. El texto subrayado en el análisis a continuación representa los cambios respecto a JCTVC-N1005. Las eliminaciones de la propuesta de JCTVC-N1005 no se enfatizan necesariamente en el análisis a continuación, pero pueden observarse comparando este análisis con el análisis de JCTVCN1005 presentado anteriormente.

50 Las variables log2Wd, o0, o1 y w0, w1 se obtienen como sigue:

- Si cldx es igual a 0 para muestras de luma, se aplica lo siguiente:

$$\text{log2Wd} = \text{luma_log2_weight_denom} + \text{shift1} \quad (8-243)$$

$$\text{w0} = \text{LumaWeightL0}[\text{refIdxL0}] \quad (8-244)$$

$$\text{w1} = \text{LumaWeightL1}[\text{refIdxL1}] \quad (8-245)$$

$$\text{o0} = \text{luma_offset_10}[\text{refIdxL0}] \quad (8-246')$$

$$\text{o1} = \text{luma_offset_11}[\text{refIdxL1}] \quad (8-247')$$

- De lo contrario (cldx no es igual a 0 para las muestras de croma), se aplica lo siguiente:

$$\text{log2Wd} = \text{ChromaLog2WeightDenom} + \text{shift1} \quad (8-248)$$

$$w0 = \text{ChromaWeightL0}[\text{refIdxL0}][\text{cldx} - 1] \quad (8-249)$$

$$w1 = \text{ChromaWeightL1}[\text{refIdxL1}][\text{cldx} - 1] \quad (8-250)$$

$$5 \quad o0 = \text{ChromaOffsetL0}[\text{refIdxL0}][\text{cldx} - 1] \quad (8-251')$$

$$o1 = \text{ChromaOffsetL1}[\text{refIdxL1}][\text{cldx} - 1] \quad (8-252')$$

JCTVC-N1005 especifica las fórmulas (8-246), (8-247), (8-251) y (8-252) como sigue:

$$10 \quad o0 = \text{luma_offset_l0}[\text{refIdxL0}] * (1 \ll (\text{bitDepth} - 8)) \quad (8-246)$$

$$o1 = \text{luma_offset_l1}[\text{refIdxL1}] * (1 \ll (\text{bitDepth} - 8)) \quad (8-247)$$

$$15 \quad o0 = \text{ChromaOffsetL0}[\text{refIdxL0}][\text{cldx} - 1] * (1 \ll (\text{bitDepth} - 8)) \quad (8-251)$$

$$o1 = \text{ChromaOffsetL1}[\text{refIdxL1}][\text{cldx} - 1] * (1 \ll (\text{bitDepth} - 8)) \quad (8-252)$$

20 Como puede verse por medio de la comparación, en las fórmulas (8-246'), (8-247'), (8-251') y (8-252') de esta divulgación, las variables o0 y o1 pueden calcularse sin multiplicar por $(1 \ll (\text{bitDepth} - 8))$, como se especifica en JCTVC-N1005. De forma similar, las variables o0 y o1 pueden calcularse sin calcular $(1 \ll (\text{la profundidad de bits menos } 8))$, donde "<<" representa el operador de desplazamiento a la izquierda en bits.

25 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para codificar datos indicativos de si la predicción ponderada de alta precisión está habilitada, por ejemplo, para una secuencia de imágenes. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para codificar un elemento de sintaxis de un conjunto de parámetros de secuencia (SPS), donde el valor del elemento de sintaxis indica si está habilitada la predicción ponderada de alta precisión (es decir, una profundidad de bits alta) para los datos de vídeo. La tabla 2 a continuación presenta la sintaxis que puede usarse para dichos datos indicativos de si está habilitada la predicción ponderada de alta precisión. De acuerdo con el ejemplo de la FIG. 2, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar un indicador para indicar si la predicción ponderada de alta precisión se usa en una sintaxis de SPS RBSP. El indicador también puede señalarse en otro cuerpo de sintaxis de alto nivel, tal como un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de imagen (PPS), un encabezado de fragmento, un encabezado de bloque o similar. En el ejemplo de la tabla 2, el texto subrayado representa adiciones respecto a la sintaxis de HEVC y JCTVC-N1005.

TABLA 2

seq_parameter_set_rbsp() {	Descriptor
...	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
if (BitDepth _Y > 10 BitDepth _C > 10)	
<u>use_high_precision_weighted_prediction_flag</u>	u(1)
log2_max_pic_order_cnt_lsb_minus4	ue(v)

40 La semántica para el use_high_precision_weighted_prediction_flag puede definirse como sigue:

use_high_precision_weighted_prediction_flag igual a 1 especifica que se usa la predicción ponderada explícita de alta precisión en esta divulgación. use_high_precision_weighted_prediction_flag igual a 0 especifica que no se usa la predicción ponderada explícita de alta precisión en esta divulgación y se usa el borrador de ampliación de rango de HEVC (según JCTVC-N1005). Cuando no está presente, se deduce que es 0.

45 Como puede verse en el ejemplo de la tabla 2, si la profundidad de bits para la luminancia (luma o 'Y') es mayor que 10 o si la profundidad de bits para la croma (croma o 'C') es mayor que 10, el conjunto de parámetros de secuencia puede incluir el elemento de sintaxis use_high_precision_weighted_prediction_flag, que puede indicar si se usa la predicción ponderada de alta precisión (es decir, una profundidad de bits alta). De este modo, una "profundidad de bits alta" para un proceso de predicción ponderada puede corresponder a una profundidad de bits que es mayor que diez (10) bits.

55 Además, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para funcionar de acuerdo con las siguientes modificaciones a JCTVC-N1005. En la sección 8.5.3.3.4.3 de JCTVC-N1005, shift1 se establece igual a $\text{Max}(2, 14 - \text{bitDepth})$. En el peor de los casos, las predSamples[x][y] pueden elevarse hasta: bitDepth (profundidad de bits de entrada) + shift1 + 1 (escala de interpolación) + 1 (bit de signo de interpolación).

La tabla 3 a continuación enumera la profundidad de bits de muestra del peor de los casos, modificada para adaptarse a las técnicas de esta divulgación (donde el texto subrayado representa modificaciones a la sintaxis descrita en JCTVC-N1005):

5

TABLA 3

Profundidad de bits de entrada	desplazamiento 1	Profundidad de bits de muestra
16	2	<u>20</u>
15	2	<u>19</u>
14	2	<u>18</u>
13	2	<u>17</u>
12	2	16
11	3	16

Entonces, en el peor de los casos, en la fórmula (8-253) descrita anteriormente, $\text{predSamplesL0}[x][y] * w0$ no se implementaría por el contrario usando multiplicación de 16 bits. De acuerdo con las técnicas de esta divulgación, sin embargo, $w0$ puede estar restringido a $\text{Min}(\text{bitDepth}, 16 - \text{shift1-2})$ (bit).

10

En otro ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para usar parámetros de compensación que tienen precisión ampliada, pero que usan parámetros de ponderación que tienen la misma precisión que en JCTVC-N1005. Es decir, sólo la semántica de luma_offset_10 , luma_offset_11 , $\text{delta_chroma_offset_10}$ y $\text{delta_chroma_offset_11}$ puede cambiarse de acuerdo con los diversos ejemplos descritos anteriormente, mientras que la semántica de otros elementos de sintaxis puede mantenerse sin cambios respecto a JCTVC-N1005. De este modo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar datos de vídeo usando uno o más parámetros de compensación que tienen una precisión ampliada (por ejemplo, uno o más de luma_offset_10 , luma_offset_11 , $\text{delta_chroma_offset_10}$ y $\text{delta_chroma_offset_11}$) y parámetros de ponderación que tienen precisiones definidas en JCTVCN1005.

15

20

Los parámetros de compensación, tales como luma_offset_10 , luma_offset_11 , $\text{delta_chroma_offset_10}$ y $\text{delta_chroma_offset_11}$, pueden usarse para compensar muestras de predicción de un bloque en una cantidad particular. Dichas compensaciones pueden ser beneficiosas cuando se producen cambios globales o regionales en una imagen actual, respecto a un conjunto de muestras de referencia (por ejemplo, una imagen de referencia o una parte de la imagen de referencia). Por ejemplo, si la iluminación aumenta o disminuye de manera global o regional en una imagen actual respecto a una imagen de referencia, tal como durante una escena de fundido de entrada o fundido de salida o un destello repentino de luz tal como un impacto de rayo, puede ser beneficioso introducir compensaciones a muestras en los valores de luminancia de la imagen de referencia, en lugar de intentar identificar una imagen de referencia diferente. La introducción de la compensación puede disminuir un valor residual para un bloque de la imagen actual, lo que puede reducir la velocidad de bits de los datos de vídeo.

25

30

Mientras que JCTVC-N1005 incluía un intervalo fijo de valores posibles para dichos desplazamientos (es decir, profundidades de bits fijas para señalar dichas compensaciones), las técnicas de esta divulgación permiten ajustar dinámicamente la profundidad de bits de dichas compensaciones. Es decir, el codificador de vídeo 20 puede señalar uno o más valores representativos de la profundidad de bits para los parámetros de compensación, así como si está habilitada la profundidad de bits alta (es decir, si está habilitado un proceso de predicción ponderada de alta precisión). El descodificador de vídeo 30, de forma similar, puede usar los valores señalados representativos de la profundidad de bits para descodificar correctamente el flujo de bits. Más en particular, el descodificador de vídeo 30 puede determinar los bits del flujo de bits que se aplican a los parámetros de compensación y cuándo se comienzan a aplicar los bits a un elemento de sintaxis diferente subsiguiente. Por ejemplo, si la profundidad de bits es de 8 bits y el descodificador de vídeo 30 ha descodificado 8 bits para un parámetro de compensación, el descodificador de vídeo 30 puede determinar que un bit subsiguiente se aplica a un elemento de sintaxis diferente, y no al parámetro de compensación.

35

40

45

Permitir una profundidad de bits más alta para parámetros de compensación que las profundidades de bits de dichos parámetros especificados en JCTVCN1005 puede dar ciertas ventajas. Por ejemplo, la eficiencia de codificación (es decir, el número de bits incluidos en el flujo de bits) puede mejorarse cuando los datos de vídeo de entrada tienen una profundidad de bits relativamente alta. Sin embargo, cuando los datos de vídeo de entrada tienen una profundidad de bits relativamente baja o cuando el número de bits usados para los parámetros de compensación no da ganancias de codificación significativas (como se muestra, por ejemplo, mediante medidas de distorsión-velocidad), el codificador de vídeo 20 puede desactivar la profundidad de bits alta.

50

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una

- variedad de circuitería de codificador o decodificador adecuada, según corresponda, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), circuitería de lógica discreta, software, hardware, firmware o cualesquiera combinaciones de los mismos. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.
- La FIG. 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar las técnicas para predicción ponderada explícita de alta precisión. El codificador de vídeo 20 puede realizar intra e intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en vídeo dentro de un fotograma o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de fotogramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como predicción unidireccional (modo P) o bipredicción (modo B), pueden referirse a cualquiera de varios modos de codificación de base temporal.
- Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fotograma de vídeo que va a codificarse. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye la unidad de selección de modo 40, la memoria de imágenes de referencia 64, el sumador 50, la unidad de procesamiento de transformada 52, la unidad de cuantificación 54 y la unidad de codificación entrópica 56. A su vez, la unidad de selección de modo 40 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46 y la unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también la unidad de cuantificación inversa 58, la unidad de transformación inversa 60 y el sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la FIG. 2) para filtrar límites de bloque para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará típicamente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o posteriores al bucle) además del filtro de desbloqueo. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).
- Durante el proceso de codificación, el codificador de vídeo 20 recibe un fotograma o un fragmento de vídeo que va a codificarse. El fotograma o el fragmento puede dividirse en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan codificación interpredictiva del bloque de vídeo recibido respecto a uno o más bloques en uno o más fotogramas de referencia para proporcionar predicción temporal. La unidad de intrapredicción 46 puede de forma alternativa realizar codificación intrapredictiva del bloque de vídeo recibido respecto a uno o más bloques vecinos del mismo fotograma o fragmento que el bloque que va a codificarse para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.
- Además, la unidad de división 48 puede dividir bloques de datos de vídeo en subbloques, basado en la evaluación de esquemas de división previos en pasadas de codificación previas. Por ejemplo, la unidad de división 48 puede dividir inicialmente un fotograma o un fragmento en LCU, y dividir cada una de las LCU en sub-CU, basado en análisis de distorsión-velocidad (por ejemplo, optimización de distorsión-velocidad). La unidad de selección de modo 40 puede producir adicionalmente una estructura de datos de árbol cuaternario que indica la división de una LCU en sub-CU. Las CU de nodo de hoja del árbol cuaternario pueden incluir una o más PU y una o más TU.
- La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, basado en resultados de error, y proporciona el bloque intra o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como un fotograma de referencia. La unidad de selección de modo 40 proporciona también elementos de sintaxis, tales como vectores de movimiento, indicadores de intramodo, información de división y otra información sintáctica de este tipo, a la unidad de codificación entrópica 56.
- La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman 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 un fotograma o imagen de vídeo actual respecto a un bloque predictivo dentro de un fotograma de referencia (u otra unidad codificada) respecto al bloque actual que está codificándose dentro del fotograma actual (u otra unidad codificada). Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque que va a codificarse, en lo que respecta a diferencia de píxeles, que puede determinarse mediante suma de diferencia absoluta (SAD), suma de diferencia de cuadrados (SSD) u otras medidas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel de 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 respecto a las posiciones de píxel completo y las posiciones de píxel fraccionario y emitir un vector de movimiento con precisión de píxel fraccionario.

5 La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo de un fragmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse 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.

15 La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basado en el 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 integrarse de forma funcional, en algunos ejemplos. Luego de 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 los valores de píxel del bloque predictivo de los valores de píxel del bloque de vídeo actual que está codificándose, formando valores de diferencia de píxel, como se analiza a continuación. Por lo general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento respecto a los componentes de luma, y la unidad de compensación de movimiento 44 usa los vectores de movimiento calculados basados en los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modo 40 también puede generar elementos de sintaxis asociados a los bloques de vídeo y al fragmento de vídeo para su uso mediante el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

30 La unidad de compensación de movimiento 44 puede configurarse además para aplicar compensaciones a muestras de predicción durante la compensación de movimiento. Por ejemplo, la unidad de selección de modo 40 puede hacer que la unidad de compensación de movimiento 44 aplique compensaciones a muestras de luminancia y/o crominancia de un bloque de referencia (o a una imagen de referencia completa), lo cual puede mejorar la predicción del bloque de referencia o de la imagen de referencia para una imagen actual. De acuerdo con las técnicas de esta divulgación, la unidad de selección de modo 40 puede seleccionar compensaciones ya sea de una profundidad de bits convencional o de una profundidad de bits relativamente alta. Cuando las compensaciones son de profundidad de bits alta, el codificador de vídeo 20 puede codificar un valor para un elemento de sintaxis (por ejemplo, de un conjunto de parámetros de secuencia (SPS)) que indica que está habilitada la profundidad de bits alta.

40 Como ejemplo, el codificador de vídeo 20 puede configurarse para aplicar compensaciones a los datos de predicción de luminancia de una o más imágenes de referencia. Si la imagen actual es una imagen predicha de forma unidireccional (es decir, una imagen P), el codificador de vídeo 20 puede aplicar las compensaciones a una o más imágenes de referencia en la lista 0 de imágenes de referencia. El codificador de vídeo 20 puede señalar valores para parámetros representativos de compensaciones que van a aplicarse a una o más imágenes de referencia en la lista 0 de imágenes de referencia. Es decir, el codificador de vídeo 20 puede señalar valores para $\text{luma_offset_10}[i]$, donde i está en el intervalo de todas las imágenes de referencia en la lista 0 de imágenes de referencia. De forma similar, si la imagen actual es una imagen B, el codificador de vídeo 20 puede señalar valores para los parámetros de compensación de las imágenes de referencia tanto en la lista 0 de imágenes de referencia como en la lista 1 de imágenes de referencia, donde los parámetros de compensación para las imágenes de referencia de la lista 0 de imágenes de referencia pueden corresponder a $\text{luma_offset_10}[i]$ y los parámetros de compensación para las imágenes de referencia de la lista 1 de imágenes de referencia pueden corresponder a $\text{luma_offset_11}[i]$.

50 De forma similar, el codificador de vídeo 20 puede codificar valores para parámetros de compensación para muestras de crominancia. Por ejemplo, el codificador de vídeo 20 puede codificar valores para los parámetros $\text{delta_chroma_offset_10}[i][j]$ y, si la imagen actual es una imagen B, para los parámetros $\text{delta_chroma_offset_11}[i][j]$. De nuevo, "10" representa la lista 0 de imágenes de referencia mientras que "11" representa la lista 1 de imágenes de referencia, e "i" está en el intervalo de imágenes en la lista de imágenes de referencia respectiva. La variable 'j' para estos parámetros representa un valor basado en un índice de contexto "cldx", a saber, $\text{cldx}-1$. El valor de cldx puede ser cero (0) para muestras de luma y distinto de cero para muestras de croma.

60 El codificador de vídeo 20 puede señalar los valores para los parámetros $\text{luma_offset_10}[i]$, $\text{luma_offset_11}[i]$, $\text{delta_chroma_offset_10}[i][j]$ y $\text{delta_chroma_offset_11}[i][j]$ en una tabla de ponderación de predicción (pred_weight_table), que puede formar parte de un encabezado de fragmento.

65 Por consiguiente, la unidad de compensación de movimiento 44 puede calcular valores para las variables o_0 y o_1 de acuerdo con las fórmulas (8-246'), (8-247'), (8-251') y (8-252'), como se analizó anteriormente con respecto a la FIG. 1. La unidad de compensación de movimiento 44 puede aplicar las compensaciones a bloques de referencia determinados a partir de imágenes de referencia respectivas. Por consiguiente, cuando el codificador de vídeo 20

forma y codifica bloques residuales, el sumador 50 puede calcular los bloques residuales como diferencias de píxel a píxel (o de muestra a muestra) entre un bloque no codificado actual y el bloque de referencia formado por la unidad de compensación de movimiento 44 (donde la unidad de compensación de movimiento 44 puede haber aplicado las compensaciones al bloque de referencia). Como se analizó anteriormente, la unidad de selección de modo 40 puede hacer que la unidad de compensación de movimiento 44 pruebe diversos valores para las compensaciones y seleccione valores para los parámetros de compensación que dan las mejores características de distorsión-velocidad. Además, cuando se habilita la profundidad de bits alta, la unidad de selección de modo 40 puede probar valores para los parámetros de compensación dentro de los intervalos definidos por la profundidad de bits alta.

La unidad de intrapredicción 46 puede intrapredicir 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 para usarse 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 independientes, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usarse a partir de los modos probados.

Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de distorsión-velocidad usando un análisis de distorsión-velocidad para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tenga las mejores características de distorsión-velocidad entre los modos probados. El análisis de distorsión-velocidad determina en general una cantidad de distorsión (o error) entre un bloque codificado y un bloque no codificado original que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados para determinar el modo de intrapredicción que manifiesta el mejor valor de distorsión-velocidad para el bloque.

Después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionada 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 los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificados (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, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificados para usarse para cada uno de los contextos.

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 está codificándose. 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 de coseno discreta (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformada residuales. La unidad de procesamiento de transformada 52 puede realizar otras transformadas que son conceptualmente similares a la DCT. También podrían usarse transformadas de ondículas, transformadas de números enteros, transformadas de subbanda 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 residuales. La transformada puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformada, tal como un dominio frecuencial. 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 adicionalmente la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. El proceso de cuantificación también puede denominarse proceso de "escalado", y de este modo, los coeficientes de transformada cuantificados también pueden denominarse "coeficientes de transformada escalados". El grado de cuantificación (o escalado) puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de codificación entrópica 56 puede realizar posteriormente una exploración de la matriz incluyendo los coeficientes de transformada cuantificados.

Tras la cuantificación, la unidad de codificación entrópica 56 codifica de manera entrópica los coeficientes de transformada cuantificados explorados. 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 la sintaxis (SBAC), codificación entrópica de división de intervalos de probabilidad (PIPE) u otra técnica de codificación entrópica. En el caso de la codificación entrópica basada en el contexto, el contexto puede basarse en bloques vecinos. Tras la codificación entrópica mediante la unidad de codificación entrópica 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o archivarse para su transmisión o recuperación posterior.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican cuantificación inversa y transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de uno de los fotogramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel subentero para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede usarse por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en un fotograma de vídeo subsiguiente.

De esta manera, el codificador de vídeo 20 de la FIG. 2 representa un ejemplo de un codificador de vídeo configurado para codificar un valor para un elemento de sintaxis representativo de si se habilita una profundidad de bits alta para datos de vídeo, y cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta: codificar un valor para un elemento de sintaxis representativo de una profundidad de bits para uno o más parámetros de los datos de vídeo, codificar valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits, y codificar los datos de vídeo basado al menos en parte en los valores para los parámetros.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar técnicas para la predicción ponderada explícita de alta precisión. En el ejemplo de la FIG. 3, 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 transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación en general recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos desde la unidad de descodificación entrópica 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción basados en indicadores de modo de intrapredicción recibidos desde la unidad de descodificación entrópica 70.

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 fragmento de vídeo codificado y elementos de sintaxis asociados desde el codificador de vídeo 20. La unidad de descodificación entrópica 70 del descodificador de vídeo 30 descodifica de manera entrópica el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos de sintaxis. La unidad de descodificación entrópica 70 remite los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 72. El descodificador de vídeo 30 puede recibir los elementos de sintaxis en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basados en un modo de intrapredicción señalada y datos de bloques descodificados previamente del fotograma o la imagen actual. Cuando el fotograma de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basados en los vectores de movimiento y otros elementos de sintaxis recibidos desde la unidad de descodificación entrópica 70. Los bloques predictivos pueden producirse 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 fotogramas de referencia, la lista 0 y la lista 1, usando técnicas de construcción por defecto basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

La unidad de compensación de movimiento 72 determina información de predicción para un bloque de vídeo del fragmento de vídeo actual diseccionando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que está descodificándose. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento, y otra información para descodificar los bloques de vídeo del fragmento de vídeo actual.

La unidad de compensación de movimiento 72 también puede realizar interpolación basada en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles de subentero de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos de sintaxis

recibidos y usar los filtros de interpolación para producir bloques predictivos.

De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede descodificar elementos de sintaxis de un SPS para determinar si se habilita una profundidad de bits alta para datos de vídeo. Cuando la profundidad de bits alta está habilitada, el descodificador de vídeo 30 puede descodificar valores para parámetros de compensación que tienen profundidades de bits que se basan en un valor de profundidad de bits señalado. Es decir, el SPS puede señalar si está habilitada la profundidad de bits alta y, cuando está habilitada la profundidad de bits alta, las profundidades de bits para los parámetros de compensación de luminancia y/o crominancia.

Suponiendo que el SPS indica que está habilitada la profundidad de bits alta (por ejemplo, profundidades de bits mayores que 10 para parámetros de compensación de luminancia y/o crominancia), el descodificador de vídeo 30 puede descodificar adicionalmente valores para los parámetros de compensación. El descodificador de vídeo 30 puede descodificar un encabezado de fragmento de un fragmento que incluye una tabla de ponderación de predicción (*pred_weight_table*), tal como la tabla 1, que puede señalar los valores para los parámetros de compensación. En particular, el descodificador de vídeo 30 puede usar los valores de profundidad de bits señalados (por ejemplo, señalados de acuerdo con la tabla 2 anterior) para descodificar los valores para los parámetros de compensación. Por ejemplo, el descodificador de vídeo 30 puede usar una profundidad de bits señalada para determinar partes del flujo de bits que se aplican a un parámetro de compensación en lugar de un elemento de sintaxis subsiguiente del encabezado de fragmento.

Después de descodificar los valores para los parámetros de compensación, el descodificador de vídeo 30 puede usar los parámetros de compensación para calcular valores para las variables de compensación para muestras de luminancia y/o crominancia. Las variables de compensación pueden ser, por ejemplo, las variables *o0* y *o1* como se analizó anteriormente. El descodificador de vídeo 30 puede calcular valores para las variables *o0* y *o1* usando las fórmulas (8-246'), (8-247'), (8-251') y (8-252'), como se analizó anteriormente con respecto a la FIG. 1. Posteriormente, la unidad de compensación de movimiento 72 puede aplicar estas compensaciones a muestras de referencia respectivas para producir un bloque predicho. Es decir, la unidad de compensación de movimiento 72 puede aplicar el valor de compensación de *o0* a muestras de referencia a partir de imágenes de referencia en la lista 0 de imágenes de referencia y el valor de compensación de *o1* a muestras de referencia a partir de imágenes de referencia en la lista 1 de imágenes de referencia. De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede calcular valores para las variables *o0* y *o1* sin calcular $(1 \ll (\text{la profundidad de bits menos } 8))$ y multiplicar este valor por el valor del parámetro de compensación señalada, contrariamente a las técnicas de JCTVC-N1005.

La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, 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_Y , calculado por el descodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad de transformación inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de números enteros inversa, o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de producir bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo (que puede incluir la compensación de muestras de referencia usando compensaciones que pueden tener profundidades de bits altas, como se analizó anteriormente) para el bloque de vídeo actual basadas en los vectores de movimiento y en otros elementos de sintaxis, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformación inversa 78 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques descodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden usarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles, o de lo contrario mejorar la calidad del vídeo. Los bloques de vídeo descodificados de un fotograma o imagen dada se almacenan posteriormente en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la compensación de movimiento subsiguiente. La memoria de imágenes de referencia 82 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

De esta manera, el descodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un descodificador de vídeo configurado para descodificar un valor para un elemento de sintaxis representativo de si se habilita profundidad de bits alta para los datos de vídeo, y cuando el valor para el elemento de sintaxis indica que está habilitada la profundidad de bits alta: descodifica un valor para un elemento de sintaxis representativo de una profundidad de bits para uno o más parámetros de los datos de vídeo, descodifica valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits, y descodifica los datos de vídeo basado al menos en parte en los valores

para los parámetros.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar un bloque actual. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al codificador de vídeo 20 (FIG. 1 y 2), debe entenderse que pueden configurarse otros dispositivos para realizar un procedimiento similar al de la FIG. 4.

En este ejemplo, el codificador de vídeo 20 determina inicialmente los parámetros de ponderación (150) y los parámetros de compensación (154). Los parámetros de ponderación y/o de compensación pueden tener una profundidad de bits relativamente alta, por ejemplo, una profundidad de bits de más de diez (10) bits. Por consiguiente, el codificador de vídeo 20 puede señalar (por ejemplo, en un elemento de sintaxis de un SPS, tal como `use_high_precision_weighted_prediction_flag`) si se habilita una profundidad de bits alta (154). Además, el codificador de vídeo 20 puede señalar valores que representan las profundidades de bits para los parámetros de ponderación y/o de compensación, así como los valores de los parámetros mismos (156). Por ejemplo, el codificador de vídeo 20 puede codificar valores para los parámetros `luma_offset_10[i]`, los parámetros `luma_offset_11[i]`, los parámetros `delta_chroma_offset_10[i][j]` y los parámetros `delta_chroma_offset_11[i][j]`.

Posteriormente, el codificador de vídeo 20 puede identificar un bloque de referencia para un bloque actual (158). Por ejemplo, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento para un bloque de coincidencia en una imagen de referencia. El bloque de coincidencia, usado como bloque de referencia, puede ser el bloque que manifiesta el valor de error más bajo, por ejemplo, medido por SAD, SSD, MAD, MSD u otras medidas de error de este tipo. Posteriormente, la unidad de compensación de movimiento 44 puede modificar muestras del bloque de referencia usando los parámetros de ponderación y de compensación (160). Por ejemplo, la unidad de compensación de movimiento 44 puede calcular valores para la variable `o0` y `o1` de acuerdo con las fórmulas (8-246'), (8-247'), (8-251') y (8-252'), y de este modo, sin calcular $(1 \ll (\text{bitDepth} - 8))$. De nuevo, los parámetros de ponderación y/o de compensación pueden tener una profundidad de bits relativamente alta. Por consiguiente, la unidad de compensación de movimiento 44 puede compensar muestras del bloque de referencia usando, por ejemplo, un parámetro de compensación que tiene más de diez bits. En algunos ejemplos, el parámetro de compensación puede ser un valor de dieciséis bits. El codificador de vídeo 20 puede usar el bloque de referencia modificado como un bloque predicho para el bloque actual.

Posteriormente, el codificador de vídeo 20 puede calcular un bloque residual para el bloque actual, por ejemplo, para producir una unidad de transformación (TU) (162). Para calcular el bloque residual, el codificador de vídeo 20 puede calcular una diferencia entre el bloque no codificado original y el bloque predicho para el bloque actual. Posteriormente, el codificador de vídeo 20 puede transformar y cuantificar los coeficientes del bloque residual (164). A continuación, el codificador de vídeo 20 puede explorar los coeficientes de transformada cuantificados del bloque residual (166). Durante la exploración, o tras la exploración, el codificador de vídeo 20 puede codificar de manera entrópica los coeficientes (168). Por ejemplo, el codificador de vídeo 20 puede codificar los coeficientes usando CAVLC o CABAC. Posteriormente, el codificador de vídeo 20 puede emitir los datos codificados de manera entrópica del bloque (170).

De esta manera, el procedimiento de la FIG. 4 representa un ejemplo de un procedimiento para codificar datos de vídeo que incluye determinar si se habilita una profundidad de bits alta para datos de vídeo, y después de determinar la habilitación de la profundidad de bits alta: codificar un valor para un elemento de sintaxis que indica que está habilitada la profundidad de bits alta, codificar un valor para un elemento de sintaxis representativo de una profundidad de bits para uno o más parámetros de los datos de vídeo, codificar valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en la profundidad de bits, y codificar los datos de vídeo basado al menos en parte en los valores de los parámetros.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para descodificar un bloque actual de datos de vídeo. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al descodificador de vídeo 30 (FIG. 1 y 3), debe entenderse que pueden configurarse otros dispositivos para realizar un procedimiento similar al de la FIG. 5.

Inicialmente, el descodificador de vídeo 30 puede descodificar un valor para un elemento de sintaxis representativo de si se habilita una profundidad de bits alta para datos de vídeo (200), y de este modo, si se realiza la predicción ponderada explícita de alta precisión. Suponiendo que está habilitada la profundidad de bits alta, el descodificador de vídeo 30 puede descodificar valores que representan la profundidad de bits y parámetros de ponderación y/o de compensación (202). Por ejemplo, el descodificador de vídeo 30 puede descodificar valores para los parámetros `luma_offset_10[i]`, los parámetros `luma_offset_11[i]`, los parámetros `delta_chroma_offset_10[i][j]` y los parámetros `delta_chroma_offset_11[i][j]`. Dichos valores pueden descodificarse como parte de un encabezado de fragmento, por ejemplo, de acuerdo con la tabla 1, como se ha descrito anteriormente.

Posteriormente, el descodificador de vídeo 30 puede identificar un bloque de referencia para un bloque actual (204). Por ejemplo, el descodificador de vídeo 30 puede descodificar parámetros de movimiento para el bloque actual, por ejemplo, usando el modo de fusión o el modo de predicción de vector de movimiento avanzada (AMVP). Los

parámetros de movimiento pueden identificar uno o más de una lista de imágenes de referencia, un índice en la lista de imágenes de referencia, valores de diferencia de vector de movimiento y/o un predictor de vector de movimiento de un conjunto de predictores de vector de movimiento candidatos. Posteriormente, el descodificador de vídeo 30 puede modificar muestras del bloque de referencia usando los parámetros de ponderación y de compensación (206).

5 Por ejemplo, la unidad de compensación de movimiento 72 puede calcular valores para la variable o_0 y o_1 de acuerdo con las fórmulas (8-246'), (8-247'), (8-251') y (8-252') y, de este modo, sin calcular $(1 \ll (\text{bitdepth} - 8))$. De nuevo, los parámetros de ponderación y/o de compensación pueden tener una profundidad de bits relativamente alta. Por consiguiente, la unidad de compensación de movimiento 72 puede compensar muestras del bloque de referencia usando, por ejemplo, un parámetro de compensación que tiene más de diez bits. En algunos ejemplos, el
10 parámetro de compensación puede ser un valor de dieciséis bits. El descodificador de vídeo 30 puede usar el bloque de referencia modificado como un bloque predicho para el bloque actual.

El descodificador de vídeo 30 también puede recibir datos codificados de manera entrópica para el bloque actual, tales como datos codificados de manera entrópica para coeficientes de un bloque residual correspondiente al bloque actual (208). El descodificador de vídeo 30 puede descodificar de manera entrópica los datos codificados de manera
15 entrópica para reproducir los coeficientes del bloque residual (210). Posteriormente, el descodificador de vídeo 30 puede explorar de forma inversa los coeficientes reproducidos (212), para crear un bloque de coeficientes de transformada cuantificados. Posteriormente, el descodificador de vídeo 30 puede cuantificar de manera inversa y transformar de manera inversa los coeficientes para producir un bloque residual (214). El descodificador de vídeo 30 puede finalmente descodificar el bloque actual combinando el bloque predicho y el bloque residual (216).
20

De esta manera, el procedimiento de la FIG. 5 representa un ejemplo de un procedimiento para descodificar datos de vídeo que incluye descodificar un valor para un elemento de sintaxis representativo de si se habilita una profundidad de bits alta para datos de vídeo, cuando el valor para el elemento de sintaxis indica que está habilitada
25 la profundidad de bits alta: descodificar un valor para un elemento de sintaxis representativo de una profundidad de bits para uno o más parámetros de los datos de vídeo, descodificar valores para los parámetros de tal modo que los valores para los parámetros tengan profundidades de bits basadas en el valor para el elemento de sintaxis representativo de la profundidad de bits, y descodificar los datos de vídeo basado al menos en parte en los valores para los parámetros.
30

Debe reconocerse que dependiendo del ejemplo, pueden realizarse ciertas acciones o actividades de cualquiera de las técnicas descritas en el presente documento en una secuencia diferente, pueden añadirse, fusionarse u omitirse por completo (por ejemplo, no todas las acciones o actividades descritas son necesarias para la puesta en práctica de las técnicas). Además, en ciertos ejemplos, las acciones o actividades pueden realizarse simultáneamente, por
35 ejemplo, por medio de procesamiento de múltiples subprocesos, procesamiento por interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en o transmitirse como una o más instrucciones o código en un medio legible por ordenador y ejecutarse 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 medios de almacenamiento de datos, o
40 medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento legibles por ordenador tangibles que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.
50

A modo de ejemplo, y no de manera limitativa, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético, u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. También, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota 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 se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros
55 medios transitorios, sino que en cambio se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en el presente documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde algunos discos reproducen datos magnéticamente, mientras que otros reproducen datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del
65

alcance de los medios legibles por ordenador.

5 Las instrucciones pueden ejecutarse 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 lógicas programables por campo (FPGA) u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la
10 descodificación, o incorporarse en un códec combinado. También, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

15 Las técnicas de la presente divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar aspectos funcionales de 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 pueden combinarse en una unidad de hardware de códec o proporcionarse mediante un conjunto de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente,
20 conjuntamente con software y/o firmware adecuados.

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 un valor para un elemento de sintaxis representativo de si está habilitada la predicción ponderada que tiene una profundidad de bits señalada para datos de vídeo;

cuando el valor para el elemento de sintaxis indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada:

10 descodificar un valor para un elemento de sintaxis representativo de la profundidad de bits señalada para uno o más parámetros de los datos de vídeo, en el que el uno o más parámetros comprenden una primera lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para el parámetro luma_offset_I0[i] de la predicción de lista 0 o una segunda lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para el parámetro luma_offset_I1[i] de la predicción de lista 1;

20 descodificar valores para los parámetros de tal modo que los valores para los parámetros sean representativos de las profundidades de bits que se basan en el valor para el elemento de sintaxis representativo de la profundidad de bits señalada; y

25 descodificar los datos de vídeo basados al menos en parte en los valores para los parámetros, en el que la descodificación de los datos de vídeo comprende, al realizar un proceso de predicción ponderado para muestras de luminancia:

30 calcular un valor para una primera variable [[o0]] usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la primera lista de parámetros, un parámetro luma_offset_I0[reldxL0], sin calcular ($1 \ll (\text{la profundidad de bits señalada menos } 8)$); y

35 calcular un valor para una segunda variable [[o1]] usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la segunda lista de parámetros un parámetro luma_offset_I1[reldxL1], sin calcular ($1 \ll (\text{la profundidad de bits señalada menos } 8)$).

- 40 2. El procedimiento según la reivindicación 1, en el que la descodificación del valor para el elemento de sintaxis representativo de si está habilitada la predicción ponderada que tiene la profundidad de bits señalada comprende descodificar un valor para un elemento de sintaxis de un conjunto de parámetros de secuencia (SPS) para una secuencia de imágenes de los datos de vídeo.

- 45 3. El procedimiento según la reivindicación 1, en el que la descodificación de un valor para el uno de los parámetros de la primera lista de parámetros parámetro luma_offset_I0[reldxL0] o el parámetro luma_offset_I0[i] comprende descodificar un valor en el intervalo de -2 elevado a (la profundidad de bits señalada menos 1) a 2 elevado a (la profundidad de bits señalada menos 1) menos 1, inclusive.

4. El procedimiento según la reivindicación 1, en el que el uno o más parámetros de predicción ponderada comprenden además un parámetro delta_chroma_offset_I0[i][j] o un parámetro delta_chroma_offset_I1[i][j].

- 50 5. El procedimiento según la reivindicación 4, en el que la descodificación de un valor para el parámetro delta_chroma_offset_I0[i][j] o el parámetro delta_chroma_offset_I1[i][j] comprende descodificar un valor en el intervalo de -2 elevado a (la profundidad de bits señalada más 1) a 2 elevado a (la profundidad de bits señalada más 1) menos 1 inclusive.

- 55 6. El procedimiento, de la reivindicación 5, que comprende además:

60 obtener un valor para una variable ChromaOffsetL0[i][j] de acuerdo con la fórmula: $\text{Clip3}(-2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1), 2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1) \text{ menos } 1, (\text{delta_chroma_offset_I0}[i][j]) - ((2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1) * \text{ChromaWeightL0}[i][j]) \gg \text{ChromaLog2WeightDenom}) + 2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1))$; y

65 obtener un valor para una variable ChromaOffsetL1[i][j] de acuerdo con la fórmula: $\text{Clip3}(-2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1), 2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1) \text{ menos } 1, (\text{delta_chroma_offset_I1}[i][j]) - ((2 \text{ elevado a } (\text{la profundidad de bits señalada menos } 1) * \text{ChromaWeightL1}[i][j]) \gg \text{ChromaLog2WeightDenom}) + 2 \text{ elevado a } (\text{la profundidad de bits señalada$

menos 1))).

7. El procedimiento según la reivindicación 6, en el que la descodificación de los datos de vídeo comprende, al realizar un proceso de predicción ponderado para muestras de crominancia:

5 calcular un valor para una variable o0 usada en el proceso de predicción ponderada que es igual a una variable $\text{ChromaOffsetL0}[\text{refIdxL0}][\text{cldx} - 1]$, sin calcular $(1 \ll (\text{la profundidad de bits señalada menos } 8))$;

10 calcular un valor para una variable o1 usada en el proceso de predicción ponderada que es igual a una variable $\text{ChromaOffsetL1}[\text{refIdxL1}][\text{cldx} - 1]$, sin calcular $(1 \ll (\text{la profundidad de bits señalada menos } 8))$.

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

15 determinar la habilitación de predicción ponderada que tiene una profundidad de bits señalada para datos de vídeo;

20 después de determinar la habilitación de la predicción ponderada que tiene la profundidad de bits señalada:

codificar un valor para un elemento de sintaxis que indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada;

25 codificar un valor para un elemento de sintaxis representativo de la profundidad de bits señalada para uno o más parámetros de los datos de vídeo, en el que el uno o más parámetros comprenden una primera lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para un parámetro $\text{luma_offset_l0}[i]$ de la predicción de lista 0 o una segunda lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para el parámetro $\text{luma_offset_l1}[i]$ de la predicción de lista 1;

codificar valores para los parámetros de tal modo que los valores para los parámetros sean representativos de profundidades de bits que se basan en la profundidad de bits señalada; y

35 codificar los datos de vídeo basados al menos en parte en los valores para los parámetros, en el que la codificación de los datos de vídeo comprende, al realizar un proceso de predicción ponderada para muestras de luminancia:

40 calcular un valor para una primera variable $[[o0]]$ usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la primera lista de parámetros, un parámetro $\text{luma_offset_l0}[\text{refIdxL0}]$, sin calcular $(1 \ll (\text{la profundidad de bits señalada menos } 8))$; y

45 calcular un valor para una segunda variable $[[o1]]$ usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la segunda lista de parámetros, un parámetro $\text{luma_offset_l1}[\text{refIdxL1}]$, sin calcular $(1 \ll (\text{la profundidad de bits señalada menos } 8))$.

9. El procedimiento según la reivindicación 8, en el que la codificación del valor para el elemento de sintaxis que indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada comprende codificar un valor para un elemento de sintaxis de un conjunto de parámetros de secuencia (SPS) para una secuencia de imágenes de los datos de vídeo.

10. El procedimiento según la reivindicación 8, en el que la codificación de un valor para el uno de los parámetros de la primera lista de parámetros, parámetro $\text{luma_offset_l0}[\text{refIdxL0}]$, o el uno de los parámetros de la segunda lista de parámetros, parámetro $\text{luma_offset_l1}[\text{refIdxL0}]$, comprende codificar un valor en el intervalo de -2 elevado a $(\text{la profundidad de bits señalada menos } 1)$ a 2 elevado a $(\text{la profundidad de bits señalada menos } 1)$ menos 1, inclusive.

11. El procedimiento según la reivindicación 8, en el que el uno o más parámetros de predicción ponderada comprenden un parámetro $\text{delta_chroma_offset_l0}[i][j]$ o un parámetro $\text{delta_chroma_offset_l1}[i][j]$, y en el que la codificación de un valor para el parámetro $\text{delta_chroma_offset_l0}[i][j]$ o el parámetro $\text{delta_chroma_offset_l1}[i][j]$ comprende codificar un valor en el intervalo de -2 elevado a $(\text{la profundidad de bits señalada más } 1)$ a 2 elevado a $(\text{la profundidad de bits señalada más } 1)$ menos 1 inclusive.

12. El procedimiento según la reivindicación 11, en el que la codificación de los datos de vídeo comprende, al

realizar un proceso de predicción ponderada para muestras de crominancia:

5 calcular un valor para una variable $ChromaOffsetL0[refIdxL0][cIdx - 1]$ a partir del parámetro $\Delta chroma_offset_l0[i][j]$;

calcular un valor para una variable $o0$ usada en el proceso de predicción ponderada que es igual a la variable $ChromaOffsetL0[refIdxL0][cIdx - 1]$, sin calcular $(1 \ll (la\ profundidad\ de\ bits\ señalada\ menos\ 8))$;

10 calcular un valor para una variable $ChromaOffsetL1[refIdxL1][cIdx - 1]$ a partir del parámetro $\Delta chroma_offset_l1[i][j]$; y

15 calcular un valor para una variable $o1$ usada en el proceso de predicción ponderada que es igual a la variable $ChromaOffsetL1[refIdxL1][cIdx - 1]$, sin calcular $(1 \ll (la\ profundidad\ de\ bits\ señalada\ menos\ 8))$.

13. El procedimiento según la reivindicación 8, en el que el elemento de sintaxis representativo de si está habilitada la predicción ponderada que tiene la profundidad de bits señalada comprende `use_high_precision_weighted_prediction_flag`.

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

25 medios para codificar un valor para un elemento de sintaxis representativo de si está habilitada la predicción ponderada que tiene una profundidad de bits señalada para datos de vídeo;

30 medios para codificar un valor para un elemento de sintaxis representativo de la profundidad de bits señalada para uno o más parámetros de los datos de vídeo cuando el valor para el elemento de sintaxis indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada, en el que el uno o más parámetros comprenden una primera lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para un parámetro $luma_offset_l0[i]$ de la predicción de lista 0 o una segunda lista de parámetros que representan cada uno una compensación aditiva que va a aplicarse a un valor de predicción de luma para el parámetro $luma_offset_l1[i]$ de la predicción de lista 1;

35 medios para codificar valores para los parámetros de tal modo que los valores para los parámetros sean representativos de profundidades de bits que se basan en el valor para el elemento de sintaxis representativo de la profundidad de bits señalada cuando el valor para el elemento de sintaxis indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada; y

40 medios para codificar los datos de vídeo basados al menos en parte en los valores para los parámetros cuando el valor para el elemento de sintaxis indica que está habilitada la predicción ponderada que tiene la profundidad de bits señalada, en el que los medios para codificar los datos de vídeo comprenden:

45 medios para calcular un valor para una primera variable $[o0]$ usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la primera lista de parámetros un parámetro $luma_offset_l0[refIdxL0]$, sin calcular $(1 \ll (la\ profundidad\ de\ bits\ señalada\ menos\ 8))$; y

50 medios para calcular un valor para una segunda variable $[o1]$ usada en el proceso de predicción ponderada que es igual a un valor para uno de los parámetros de la segunda lista de parámetros un parámetro $luma_offset_l1[refIdxL1]$, sin calcular $(1 \ll (la\ profundidad\ de\ bits\ señalada\ menos\ 8))$.

15. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un procesador lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 13.

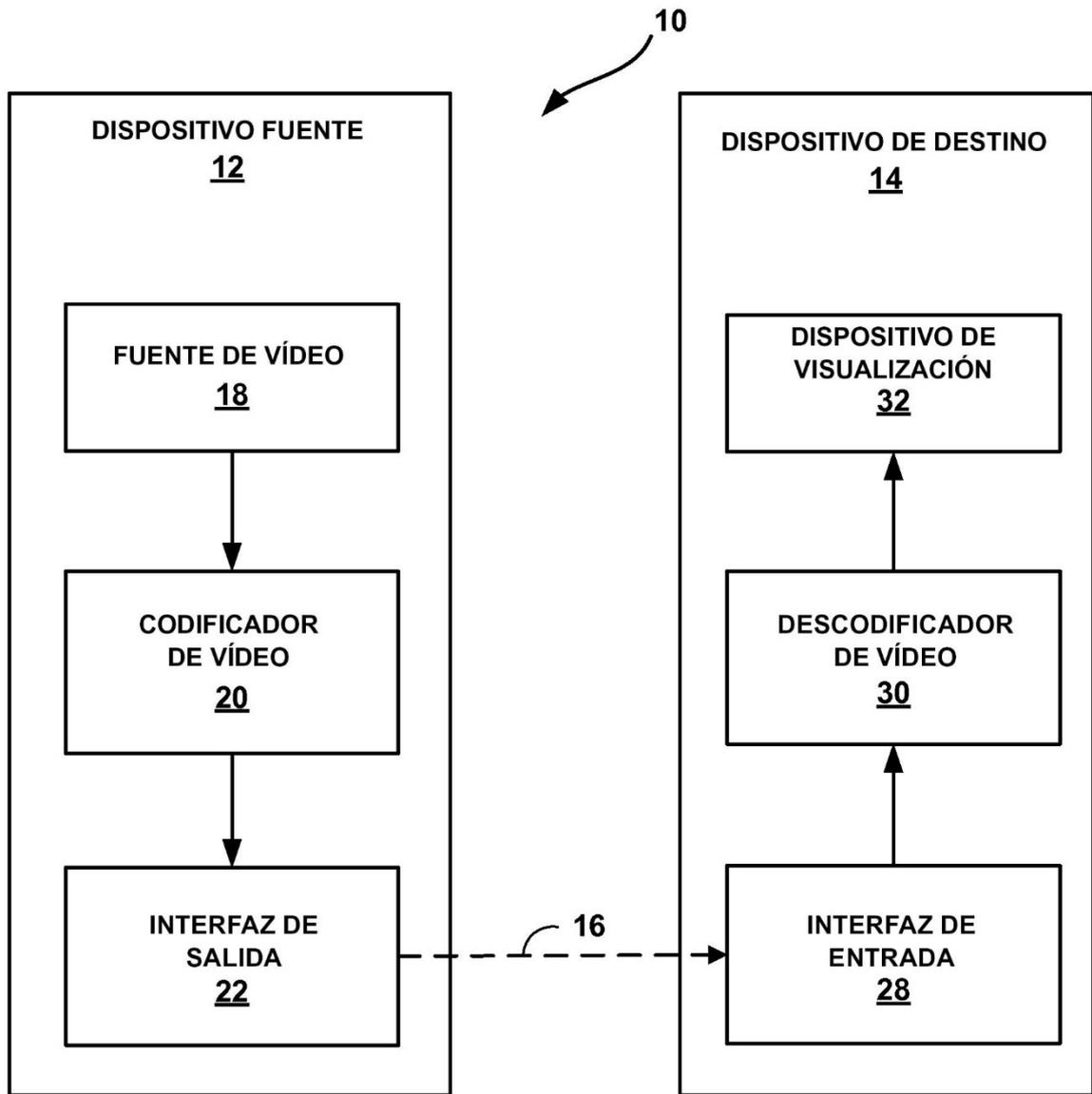


FIG. 1

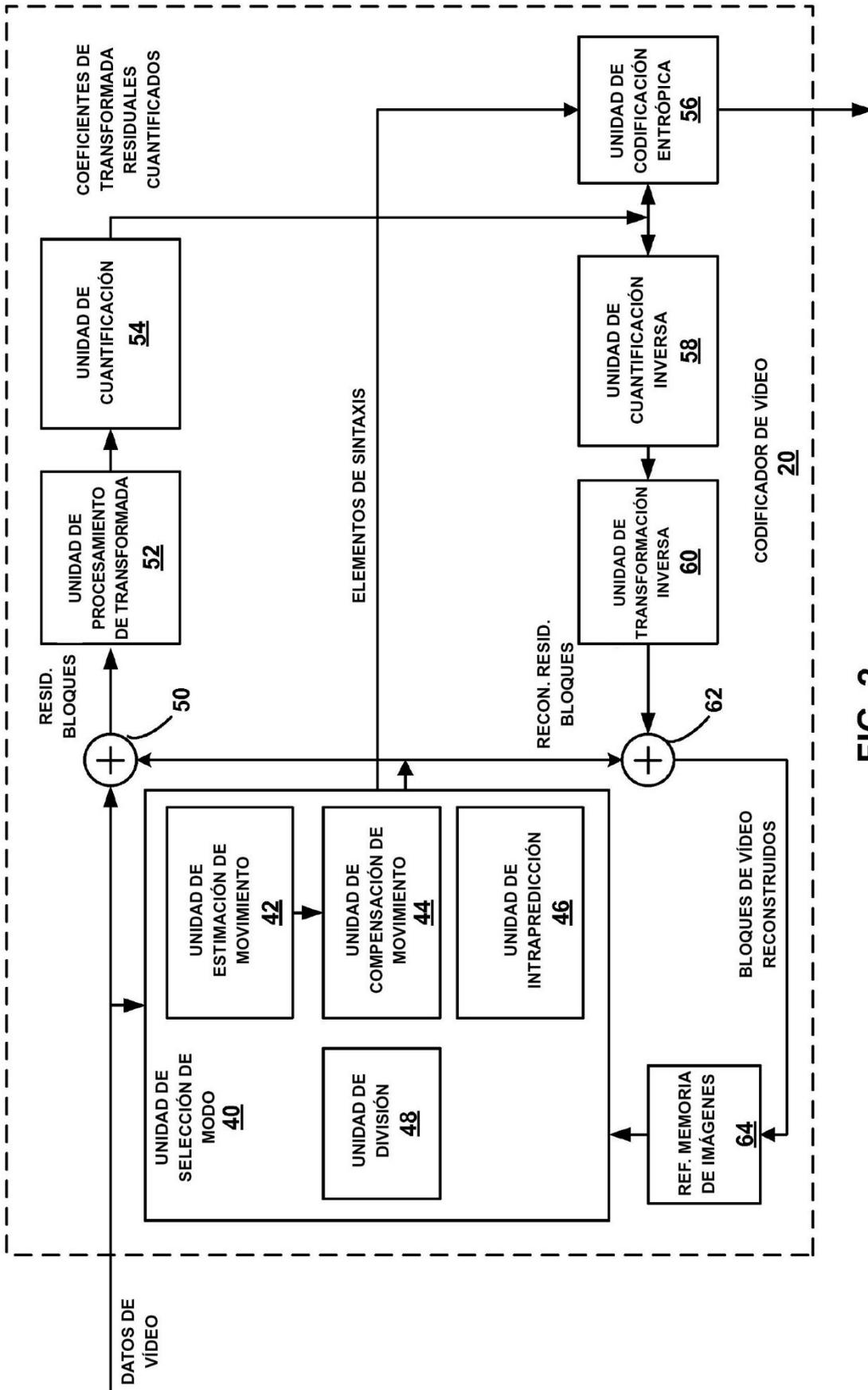


FIG. 2

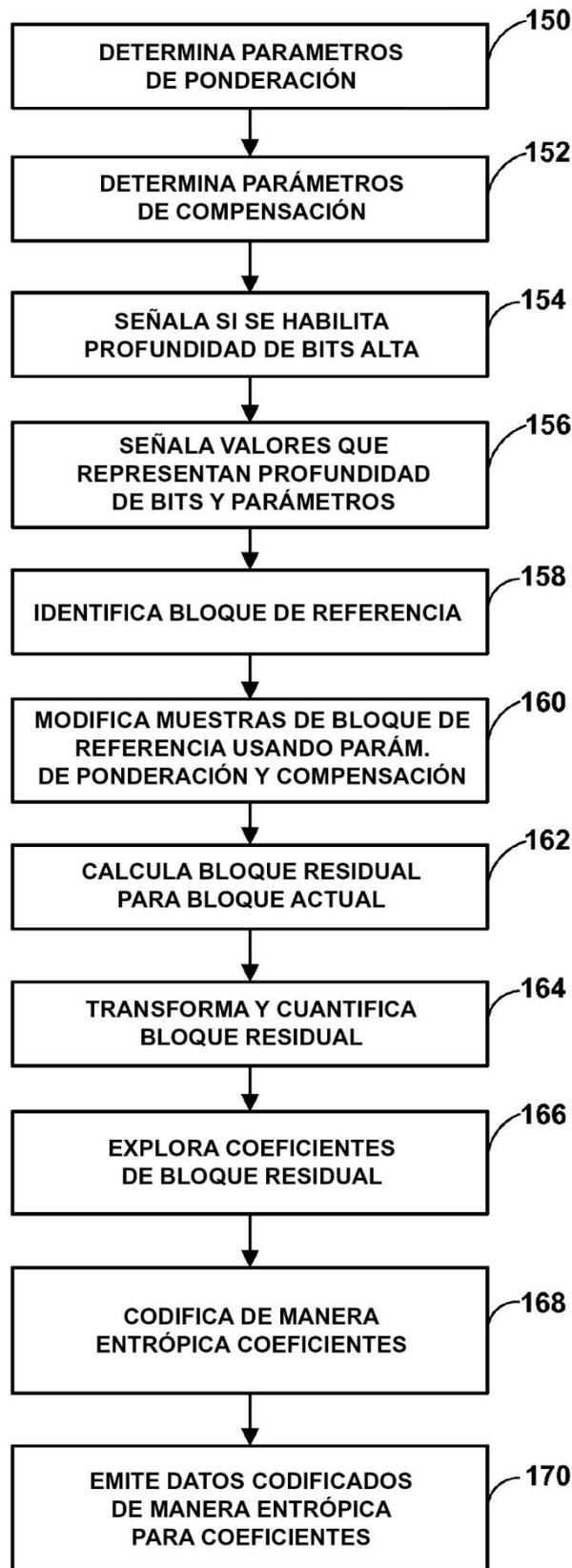


FIG. 4

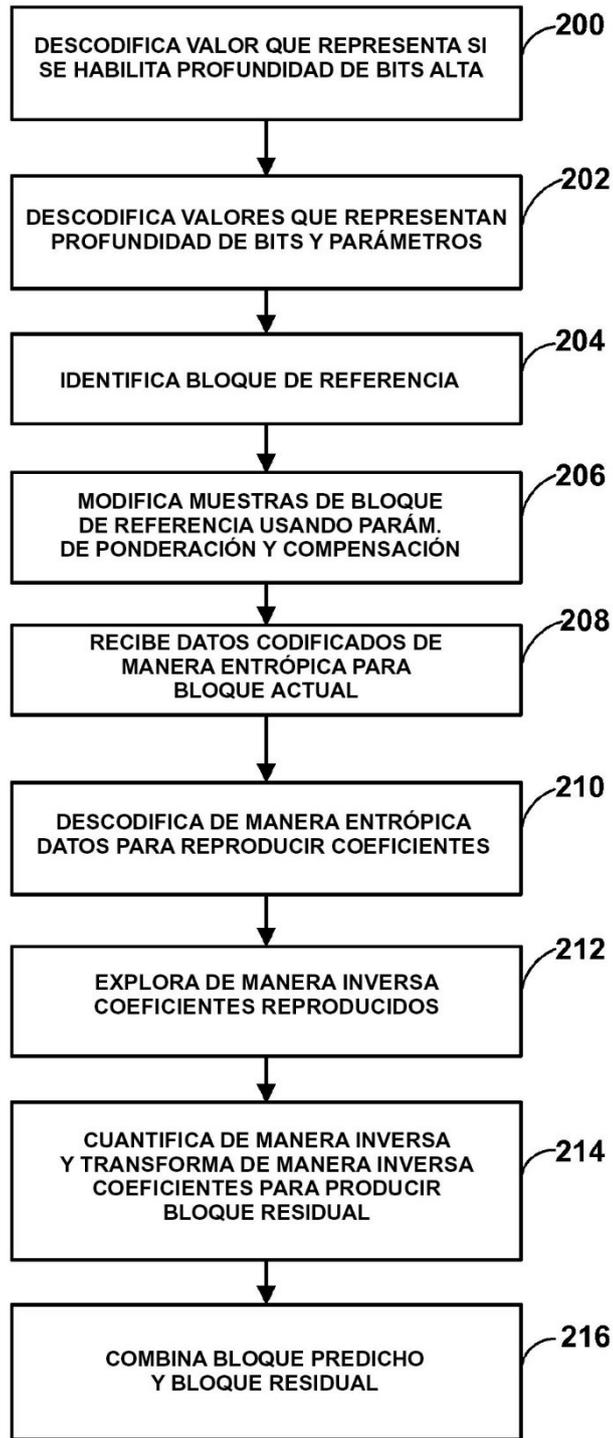


FIG. 5