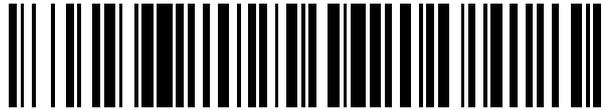


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 620 432**

51 Int. Cl.:

H04N 19/597	(2014.01)
H04N 19/70	(2014.01)
H04N 19/105	(2014.01)
H04N 19/176	(2014.01)
H04N 19/61	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **20.11.2013 PCT/US2013/070957**
- 87 Fecha y número de publicación internacional: **30.05.2014 WO2014081799**
- 96 Fecha de presentación y número de la solicitud europea: **20.11.2013 E 13802489 (8)**
- 97 Fecha y número de publicación de la concesión europea: **21.12.2016 EP 2923492**

54 Título: **Compensación de luminancia adaptativa en la codificación de vídeo tridimensional**

30 Prioridad:

20.11.2012 US 201261728739 P
08.01.2013 US 201361750321 P
19.11.2013 US 201314084347

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
28.06.2017

73 Titular/es:

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

72 Inventor/es:

CHEN, YING;
KANG, JEWON y
ZHANG, LI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 620 432 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

Compensación de luminancia adaptativa en la codificación de vídeo tridimensional

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo y, más en particular, a la codificación de vídeo tridimensional (3DVC).

10 ANTECEDENTES

15 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 videojuegos, teléfonos celulares 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 compresión 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 Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) actualmente en desarrollo y las extensiones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo.

25 Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intra-imagen) 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, una imagen o una parte de una imagen) puede ser dividida en bloques de vídeo, que también pueden ser denominados bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intra-codificado (I) de una imagen son codificados usando la predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia.

35 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que indica la ubicación de 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 intra-codificado se codifica de acuerdo con una modalidad de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, los cuales pueden cuantizarse posteriormente. Los coeficientes de transformación cuantizados, inicialmente dispuestos en una formación bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr aún más compresión.

45 La propuesta JCT3V-B0031 por Mikhail Mishurovskiy et al describe "3D-CE2.a da como resultado la codificación entre vistas con compensación de luminancia adaptativa", del 12 de octubre de 2012.

50 RESUMEN

55 En general, esta divulgación describe técnicas para la realización de la codificación de vídeo tridimensional, utilizando una extensión tridimensional de la norma H.264/Codificación de Vídeo Avanzada (AVC), conocida como 3D-AVC. La extensión tridimensional define una tecnología de codificación de vídeo para la codificación de múltiples vistas de datos de vídeo, tales como vistas de ojo izquierdo y derecho con los datos de profundidad. Cada vista puede corresponder a una perspectiva, o ángulo, diferente, donde fueron capturados los datos de vídeo correspondientes de una escena común. En el contexto de la norma 3D H.264/AVC, cada vista puede contener una vista de textura y una vista de profundidad. Una representación codificada de una vista en un caso momentáneo es un componente de vista. Un componente de vista puede contener un componente de vista en profundidad y un componente de vista en textura.

60 En la norma 3D-AVC, un codificador de vídeo puede realizar una técnica conocida como "compensación de la iluminación". Las técnicas de compensación de la iluminación ajustan o normalizan los valores de muestra de iluminación o "luma", para un bloque actual en base a las diferencias de las muestras de luma codificadas previamente de la vista actual y las muestras de luma de otra vista. Al normalizar los valores de luma entre las diferentes vistas, un codificador de vídeo puede predecir mejor los valores de las muestras similares o de luma en comparación con los valores de luma no compensados. Sin embargo, las técnicas actuales para realizar la

compensación de la iluminación en la norma 3D-AVC pueden incurrir en alta complejidad de cálculo y/o pueden ser incompatibles con las implementaciones existentes de decodificadores de la norma H.264/AVC en algunos casos. Las técnicas de esta divulgación se refieren en general a las técnicas para reducir la complejidad de cálculo y mejorar la compatibilidad de un codificador de vídeo configurado para realizar la compensación de la iluminación en la norma 3D-AVC.

La invención se define en las reivindicaciones a las que se dirige ahora la referencia.

Los detalles de uno o más aspectos de la divulgación se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas de las técnicas descritas en esta divulgación serán evidentes a partir de la descripción y 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 a modo de ejemplo de codificación y decodificación de vídeo 10, que puede configurarse para realizar la compensación de iluminación adaptativa, de acuerdo con las técnicas de esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo 20 a modo de ejemplo, que puede estar configurado para realizar la compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo 30 a modo de ejemplo, que puede configurarse para realizar la compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

La FIG. 4 es un diagrama conceptual que ilustra un patrón de predicción de la norma 3D-AVC a modo de ejemplo.

La FIG. 5 es un diagrama conceptual que ilustra la relación entre un bloque de una trama de referencia predicha entre vistas, y un bloque actual de una trama decodificada.

Las FIGs. 6A a 6D son diagramas conceptuales que ilustran ejemplos de técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

Las FIGs. 7A a 7B son diagramas conceptuales que ilustran ejemplos de técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra una operación a modo de ejemplo de un dispositivo de codificación de vídeo configurado para realizar técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

De acuerdo con ciertos sistemas de codificación de vídeo, la estimación de movimiento y la compensación de movimiento se pueden utilizar para reducir la redundancia temporal en una secuencia de vídeo, con el fin de lograr la compresión de datos. En este caso, puede generarse un vector de movimiento que identifica un bloque predictivo de datos de vídeo, por ejemplo, un bloque de otra imagen o fragmento de vídeo, que se puede utilizar para predecir los valores del bloque de vídeo actual que se está codificando. Los valores del bloque de vídeo predictivo se restan de los valores del bloque de vídeo actual para producir un bloque de datos residuales. La información de movimiento (por ejemplo, un vector de movimiento, índices de vectores de movimiento, direcciones de predicción u otra información) se comunica desde un codificador de vídeo a un decodificador de vídeo, junto con los datos residuales. El decodificador puede localizar el mismo bloque predictivo (en base al vector de movimiento) y reconstruir el bloque de vídeo codificado mediante la combinación de los datos residuales con los datos del bloque predictivo.

La norma tridimensional H.264/AVC (3D-AVC) es un proceso de codificación de vídeo para la codificación de varias vistas de datos de vídeo, para producir un efecto tridimensional, por ejemplo, usando una vista del ojo izquierdo y una vista del ojo derecho. En general, cada vista se corresponde con una perspectiva, o ángulo, diferente, donde fueron capturados (o generados) los datos de vídeo correspondientes de una escena común. Una extensión 3D-AVC a la norma ITU-T H.264/AVC está actualmente en fase de desarrollo. Un borrador de trabajo de una enmienda de la norma H.264/AVC, para añadir la extensión 3DVC, se describe en "3D-AVC Texto Borrador 5", ISO/IEC/JTC1/SC29/WG11/JCT3V-C1002, Ginebra, Suiza, con fecha de enero de 2013 ("3D-AVC WD 5"). La extensión 3D-AVC define técnicas para la codificación de vistas, para prestar soporte a la visualización de datos de vídeo tridimensional.

Por ejemplo, en el vídeo tridimensional, dos vistas (por ejemplo, las vistas de los ojos izquierdo y derecho de un

observador humano) pueden exhibirse simultáneamente, o casi simultáneamente, utilizando diferentes polarizaciones de luz, y un espectador pueden usar gafas polarizadas pasivas de tal manera que cada uno de los ojos del espectador reciba una respectiva de las vistas. De forma alternativa, el espectador puede llevar gafas activas que obturan cada ojo de forma independiente, y una pantalla puede alternar rápidamente entre las imágenes de cada ojo de forma sincronizada con las gafas.

Si bien cada vista (por ejemplo, las vistas de los ojos izquierdo y derecho) puede codificarse de forma individual, en 3D-AVC, una de las vistas se reconstruye a partir de la otra vista usando un componente de profundidad de una vista de referencia. Para ilustrar, un mapa de profundidad de una imagen particular de una vista (donde esta imagen particular de una vista puede ser denominada un "componente de vista" de la vista) puede calcularse como una diferencia entre la vista del ojo izquierdo y la vista del ojo derecho. El codificador puede codificar la vista del ojo izquierdo, por ejemplo, como un denominado "componente de textura" del componente de vista y el mapa de profundidad puede ser codificado como un denominado "componente de profundidad" del componente de vista.

El decodificador puede entonces decodificar el componente de textura del componente de vista y el componente de profundidad del componente de vista, y reconstruir la vista del ojo derecho a partir del componente de textura (que representa la vista del ojo izquierdo) utilizando el componente de profundidad. Mediante la codificación solamente de una vista y un mapa de profundidad correspondiente de esta manera, 3D-AVC puede codificar más eficazmente la vista, tanto del ojo izquierdo como del ojo derecho, en comparación con la codificación, tanto de la vista del ojo izquierdo como de la vista del ojo derecho, independientemente como vistas por separado de los datos de 3D-AVC.

3D-AVC también aprovecha las dependencias entre componentes, entre los componentes de textura y de profundidad de vídeo tridimensional. Para aprovechar estas interdependencias, 3D-AVC introduce la codificación conjunta de datos de textura y profundidad. En 3D-AVC, la vista de base está asociada con un mapa de profundidad, que indica la profundidad de las diversas regiones de una trama de datos de vídeo. Para aprovechar las dependencias entre componentes, el componente de profundidad de la vista de base se codifica con información extraída de la vista de la textura codificada. Por otra parte, el orden de codificación de los componentes de vista de la textura y de los componentes de vista de profundidad es flexible. Las vistas dependientes de la textura se codifican con información disponible de la profundidad asociada, que se codifica antes de la textura en una configuración de codificación de profundidad en primer lugar, mientras que las vistas de textura se codifican antes de la profundidad en una configuración de codificación de textura en primer lugar.

La norma 3D-AVC se basa en la norma de codificación de vídeo H.264/AVC, y un codificador de vídeo compatible con H.264/AVC puede ser capaz de codificar una vista base (por ejemplo, vistas que corresponden a un ojo) de un flujo de vídeo de 3D-AVC, pero puede no ser capaz de codificar otras vistas o componentes de profundidad de los datos de vídeo. Como parte del apoyo a la norma H.264/AVC subyacente, un codificador de vídeo compatible con 3D-AVC puede llevar a cabo la inter-predicción para predecir los componentes de luma (brillo) y croma (color) de un bloque de datos de vídeo. Durante la inter-predicción, el codificador de vídeo determina un bloque predictivo basado en un bloque de referencia codificado previamente a partir de una trama diferente que el bloque actual.

3D-AVC extiende el concepto de inter-predicción más allá al permitir la predicción entre vistas, en la que un codificador de vídeo predice valores de luma o de croma para el bloque actual, en base a los valores de un bloque de referencia, a partir de un bloque previo de una vista diferente. Sin embargo, debido a una serie de factores diferentes, los valores de luminancia (luma) de los bloques correspondientes o similares pueden variar entre las diferentes vistas. Cuando se realiza la predicción entre vistas, un codificador de vídeo de 3D-AVC (por ejemplo, un codificador o un decodificador) puede realizar la compensación de luminancia con el fin de ajustarse a esta variación de luminancia entre vistas. El proceso de realizar la compensación de luminancia puede ser denominado "compensación de luminancia adaptativa" (ALC), y también denominado "compensación de la iluminación".

Al realizar la ALC, un codificador de vídeo de 3D-AVC determina muestras de luma de los bloques contiguos superior e izquierdo en la misma trama que el bloque actual, y muestras de luma de los bloques contiguos superior e izquierdo en relación con el bloque de referencia desde otra vista. El bloque de referencia puede ser identificado por un vector de disparidad para el bloque actual. El codificador de vídeo compara las muestras de luma contiguas con las muestras de luma del bloque actual, y si los valores de las muestras varían en demasía, ajusta las muestras de luma predictivas del bloque de referencia en base a una diferencia ponderada entre los conjuntos de muestras contiguas y las muestras de luma predictivas para el bloque actual.

En 3D-AVC, al igual que con la norma H.264/AVC, una imagen de vídeo se divide en macro-bloques. Cada macro-bloque puede ser adicionalmente dividido en particiones de macro-bloque, y esas particiones de macro-bloque se pueden dividir adicionalmente en sub-particiones. Al realizar la ALC, un codificador de vídeo compatible puede llevar a cabo la compensación de movimiento para cada sub-partición de un macro-bloque, para determinar las muestras contiguas, izquierda y superior, de luma entre vistas para la ALC. En algunos casos, las sub-particiones de macro-bloque y las particiones de macro-bloque pueden tener el mismo vector exacto de movimiento, lo cual indica un bloque de referencia predictivo entre vistas de las muestras de luminancia. La realización de la compensación de movimiento varias veces, incluso cuando cada sub-partición tiene el mismo vector de movimiento, puede ser costosa en términos de cálculo. Además, los tamaños de bloque para llevar a cabo la compensación de movimiento

conjuntamente con la ALC pueden ser incompatibles con las implementaciones de decodificadores de vídeo existentes de H.264/AVC, para llevar a cabo la compensación de movimiento.

5 Las técnicas a modo de ejemplo de esta divulgación pueden utilizarse para reducir la complejidad de cálculo para realizar la compensación de la iluminación. Por ejemplo, de acuerdo con las técnicas de esta divulgación, un codificador de vídeo (por ejemplo, un codificador de vídeo o un decodificador de vídeo) puede obtener un único conjunto de ponderaciones en base a los mismos conjuntos de muestras contiguas, izquierdas y superiores, de luma entre vistas, para cada partición o sub-partición de un macro-bloque, y utilizar el mismo conjunto de ponderaciones al realizar la ALC en una partición de macro-bloque o una sub-partición. En algunos ejemplos, un codificador de vídeo también puede estar configurado para reducir o redondear de la precisión del vector de movimiento que indica la posición del bloque de referencia entre vistas, en relación con el bloque actual, con el fin de reducir la complejidad de cálculo de la realización de la compensación de movimiento en un sub-pel durante la ALC. En algunos otros ejemplos, un codificador de vídeo de acuerdo con las técnicas de esta divulgación también puede estar configurado para seleccionar las regiones de muestra contiguas, superior y/o izquierda, de luma que tengan tamaños reducidos o diferentes de muestras de luma al realizar la ALC.

La siguiente descripción debería entenderse dentro del contexto de 3D-AVC, donde una referencia a 3D-AVC se entiende que hace referencia a H.264/AVC. Es decir, dado que la 3D-AVC es una extensión para H.264, y que 3D-AVC es una extensión adicional de H.264/AVC, la 3D-AVC incorpora o, de otro modo, puede considerarse que "hereda", todos los aspectos de H.264/AVC. En otras palabras, en algunos ejemplos, las técnicas pueden proveer la generación de un flujo de bits de 3D-AVC que sea retro-compatible con la norma H.264/AVC (o, en otras palabras, capaz de ser decodificado por un decodificador de vídeo que no dé soporte a 3D-AVC pero que sí dé soporte a H.264/AVC). Si bien cada una de las siguientes técnicas se describe en el contexto de 3D-AVC, estas técnicas pueden extenderse en algunos casos a otras formas de codificación de datos de vídeo tridimensionales que tienen tanto componentes de vista de textura como componentes de vista de profundidad. Por ejemplo, las técnicas de esta divulgación pueden ser aplicadas en el contexto de las extensiones tridimensionales a otras normas de codificación de vídeo, tales como la norma de codificación de vídeo de alta eficacia (HEVC), por ejemplo, 3D-HEVC.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 a modo de ejemplo, que puede configurarse para realizar la compensación de iluminación adaptativa de acuerdo con las técnicas de esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a decodificar en un momento posterior por parte de un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, ordenadores de pizarra, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de flujos de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar, mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real, o casi tiempo real.

Los datos de vídeo codificados pueden ser modulados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El medio de comunicación puede incluir routers, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

En algunos ejemplos, pueden emitirse datos codificados desde el transmisor 24 del dispositivo de origen 24 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento por medio del receptor 26 del dispositivo de destino 14. El dispositivo de almacenamiento puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco rígido, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios adecuados cualesquiera de almacenamiento digital, 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 los datos de vídeo codificados, generados por el dispositivo de origen 12.

El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento, mediante 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. Los servidores de ficheros a modo de ejemplo incluyen un servidor de Internet (por ejemplo, para una sede en la Red), un servidor del FTP, dispositivos de almacenamiento conectados a la red (NAS) o un controlador de disco local. El dispositivo de destino 5 14 puede acceder a los datos de vídeo codificado a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado, 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 ambas. 10

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de 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, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para dar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como el flujo de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía. 15 20

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20, una unidad de encapsulación 21 y una interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, la unidad de desencapsulación 29, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 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 32. 25 30

El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas de esta divulgación relacionadas con la ALC pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, por lo general, las técnicas de esta divulgación se llevan a cabo mediante un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador/decodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también pueden llevarse a cabo mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación, donde el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y de decodificación de vídeo. Por lo tanto, el sistema 10 puede dar soporte a una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para el flujo continuo de vídeo, la reproducción de vídeo, la radiodifusión de vídeo o la videotelefonía. 35 40

El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo grabado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como una alternativa adicional, el origen de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 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 la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo grabado, pregrabado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede ser entonces emitida por la interfaz de salida 22 en un medio legible por ordenador 16. 45 50 55

El origen de vídeo 18, en general, puede proporcionar una pluralidad de vistas de una escena al codificador de vídeo 20, tales como una vista del ojo izquierdo y del derecho, correspondiente a vídeo tridimensional, para el codificador de vídeo 20. El origen de vídeo 18 también puede proporcionar información indicativa de las ubicaciones de perspectivas de la cámara para las vistas. 60

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una emisión inalámbrica o una transmisión de red cableada, 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 otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De manera similar, un dispositivo informático de una 65

instalación de producción de un medio, tal como una instalación de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluya uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

5 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 sintáctica definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). La unidad de desencapsulación 29 del dispositivo de destino 14 puede representar una unidad que desencapsula mensajes de SEI de un flujo de bits (o un subconjunto de un flujo de bits denominado un "punto de operación" en el contexto de 3D-AVC). La unidad de desencapsulación 29 puede realizar operaciones en un orden inverso a las realizadas por unidad de encapsulación 29, para desencapsular datos del flujo de bits codificado y encapsular, tales como mensajes de SEI. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario y puede comprender cualquiera entre una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

20 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre una amplia variedad de circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno, entre el codificador de vídeo 20 y el decodificador de vídeo 30, puede estar incluido en uno o más codificadores o decodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/decodificador (CODEC) 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 celular.

30 Aunque no se muestra en la FIG. 1, en algunos aspectos, cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede estar integrado con un codificador y decodificador de audio, y puede incluir unidades adecuadas de multiplexado y demultiplexado, u otro hardware y software, para gestionar la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. Si procede, las unidades de multiplexado y demultiplexado pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

35 En el ejemplo mostrado en la FIG. 1, el sistema 10 incluye también el servidor / red de suministro de contenido 34 que tiene el router 36. En algunos ejemplos, el dispositivo de origen 12 puede comunicarse con el servidor / la red de suministro de contenido 34 mediante una diversidad de medios de transmisión o de almacenamiento, inalámbricos y/o cableados, como se ha descrito anteriormente. Además, aunque se muestra por separado en el ejemplo de la FIG. 1, en algunos ejemplos, el dispositivo de origen 12 y el servidor / la red de suministro de contenido 34 comprenden el mismo dispositivo. El servidor / la red de suministro de contenido 34 puede almacenar una o más versiones de datos de vídeo codificados (desde el codificador de vídeo 20 del dispositivo de origen 12), y puede hacer que dichos datos de vídeo codificados estén disponibles para el acceso por parte del dispositivo de destino 14 y el decodificador de vídeo 30. En algunos ejemplos, el router 36 puede ser responsable de proporcionar datos de vídeo codificados al dispositivo de destino 14 en un formato solicitado.

50 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma ITU-T H.264, de forma alternativa denominada MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones de dichas normas, incluyendo la extensión MVC (codificación de vídeo de múltiples vistas) y/o la extensión 3D-AVC para la H.264. 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 la MPEG-2, la ITU-T H.263 y la HEVC (ITU-T H.265).

55 La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo de la UIT-T (VCEG), junto el Grupo de Expertos en Películas 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 ser aplicadas a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la Recomendación ITU-T H.264, Codificación de Vídeo Avanzada, para los servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T, y con fecha de marzo de 2005, que se puede denominar en el presente documento norma H.264 o memoria descriptiva H.264, o la norma o memoria descriptiva H.264/AVC. La norma H.264/AVC incluye extensiones de codificación de vídeo ajustable a escala (SVC) y extensiones de MVC. Además, hay un desarrollo adicional para proporcionar una extensión de 3D-AVC, haciendo uso de la codificación de datos de textura y datos de mapas de profundidad en diferentes resoluciones espaciales. El Equipo de Vídeo Conjunto (JVT) continúa trabajando en extensiones para H.264/MPEG-4 AVC. Si bien se han descrito en el contexto de 3D-AVC, las técnicas descritas en la presente divulgación pueden implementarse con respecto a otros algoritmos de codificación de vídeo, capaces de codificar y decodificar vídeo tridimensional que implica ambos componentes de textura y

profundidad.

Además, el codificador de vídeo 20 puede codificar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en imágenes y datos sintácticos basados en GOP, que el decodificador de vídeo 30 puede recibir finalmente, por ejemplo, en una cabecera de imagen, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un cierto número de imágenes en el GOP respectivo, y los datos sintácticos de imagen pueden indicar una modalidad de codificación/predicción usada para codificar la imagen correspondiente.

En algunos ejemplos, el codificador de vídeo 20 puede generar y el decodificador de vídeo 30 puede recibir ciertos conjuntos de parámetros, que pueden ser usados al decodificar datos de vídeo. Por ejemplo, los conjuntos de parámetros pueden contener información de cabecera a nivel de secuencia (en conjuntos de parámetros de secuencia (SPS)) e información de cabecera a nivel de imagen, que cambia con poca frecuencia (en conjuntos de parámetros de imagen (PPS)). Con los conjuntos de parámetros (por ejemplo, PPS y SPS), la información que cambia infrecuentemente no tiene por qué ser repetida para cada secuencia (por ejemplo, secuencia de imágenes) o imagen; por lo tanto, la eficacia de codificación puede mejorarse. Además, el uso de conjuntos de parámetros puede permitir la transmisión fuera de banda de la información de cabecera importante, evitando la necesidad de transmisiones redundantes, para la capacidad de recuperación de errores. En los ejemplos de transmisión fuera de banda, las unidades de NAL de conjuntos de parámetros se pueden transmitir por un canal diferente al de otras unidades de NAL, tales como las unidades de NAL de información de mejora suplementaria (SEI).

Las unidades de NAL de SEI (denominadas mensajes de SEI) pueden contener información que no es necesaria para la decodificación de las muestras de imágenes codificadas a partir de las unidades de NAL de la capa de codificación de vídeo (VCL), pero pueden ayudar en los procesos relacionados con la decodificación, la visualización, la capacidad de recuperación de errores y otros fines. Los mensajes de SEI pueden estar contenidos en las unidades de NAL no de VCL. Los mensajes de SEI pueden estar incluidos en la parte normativa de algunas memorias descriptivas de normas y, por lo tanto, no siempre son obligatorios para la implementación de un decodificador compatible con normas. Los mensajes de SEI pueden ser mensajes de SEI a nivel de secuencia o mensajes de SEI a nivel de imagen. Parte de la información a nivel de secuencia puede estar contenida en los mensajes de SEI, tales como los mensajes de SEI de información de ajustabilidad a escala en el ejemplo de la SVC y los mensajes de SEI de información de ajustabilidad a escala de vistas en la MVC. Los mensajes de SEI de información de ajustabilidad a escala también pueden proporcionar información a nivel de secuencia para la extensión 3D-AVC para el H.264.

En algunos ejemplos, el codificador de vídeo 20 puede codificar un flujo de bits de 3D-AVC que se ajusta a la extensión 3D-AVC de la norma H.264, que puede incluir datos de textura y datos de profundidad en diferentes resoluciones espaciales y, específicamente, datos de profundidad a una resolución más baja en comparación con la resolución de datos de textura. De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar configurados para realizar la compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación.

Como se ha descrito anteriormente, ciertas técnicas para realizar la compensación de luminancia adaptativa pueden dar lugar a una alta complejidad de cálculo y/o a incompatibilidad con los decodificadores de vídeo existentes de la norma H.264/AVC. En algunas implementaciones de la ALC, que pueden tener una complejidad de cálculo relativamente más alta, en comparación con las técnicas de esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden llevar a cabo la compensación de movimiento una vez para cada sub-partición de macro-bloque. Para reducir la complejidad de cálculo de la realización de la ALC, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden realizar la compensación de movimiento una sola vez para cada partición de macro-bloque.

Por ejemplo, el codificador de vídeo 20 y el decodificador 30 de vídeo pueden configurarse para obtener las ponderaciones de predicción para la compensación de iluminación de las muestras de luminancia (luma) de una partición de bloque de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación para la partición de bloques de vídeo, calcular un bloque predicho para la partición del bloque de vídeo utilizando las ponderaciones de predicción que utilizan la compensación de la iluminación, y codificar la partición del bloque de vídeo usando el bloque predicho.

La partición del bloque de vídeo puede comprender una partición de un macro-bloque. Por ejemplo, la partición del bloque de vídeo puede comprender un tamaño de 8x16, 16x8, 8x8 u otro tamaño de la partición del macro-bloque. En lugar de determinar las ponderaciones de predicción para cada bloque de tamaño 4x4 de la partición del bloque de vídeo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar las ponderaciones de predicción para la compensación de iluminación solo una vez para la partición del bloque de vídeo. Además, si el macro-bloque incluye una pluralidad (por ejemplo, dos o más) de particiones de bloques de vídeo, cada una de las particiones puede tener un conjunto común de ponderaciones de predicción, independientemente de los tamaños de

transformación para el macro-bloque. El tamaño de transformación puede ser, por ejemplo, 4x4. En otro ejemplo, un macro-bloque contiene un único conjunto de ponderaciones de predicción, obtenido una vez para el macro-bloque, independientemente de una partición de macro-bloque y los tamaños de transformación.

5 Como se ha analizado anteriormente, la compensación de la iluminación puede utilizarse en el cálculo de un bloque predicho para una partición de bloques de vídeo. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar un bloque de referencia entre vistas para la partición del bloque de vídeo utilizando, por ejemplo, la compensación de movimiento de disparidad. Además, utilizando el conjunto común de ponderaciones de predicción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden modificar los valores del bloque de referencia entre vistas, realizando la compensación de la iluminación. A continuación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar la partición del bloque de vídeo usando el bloque predicho que se sometió a la compensación de iluminación. Por ejemplo, el codificador de vídeo 20 puede calcular los datos residuales para la partición del bloque de vídeo, transformar y cuantizar los datos residuales, y codificar por entropía los datos residuales cuantizados y transformados. El decodificador de vídeo 30, de manera similar, puede decodificar por entropía datos residuales transformados y cuantizados, cuantizar inversamente y transformar inversamente los datos residuales para reproducir los datos residuales en el dominio espacial, y combinar los datos residuales con los datos predichos para la partición del bloque de vídeo, para reproducir (es decir, decodificar) la partición del bloque de vídeo. El codificador de vídeo 20 o el decodificador de vídeo 30 pueden llevar a cabo la compensación de luminancia en la compensación de movimiento en el proceso de codificación y decodificación.

20 En general, un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, que es compatible con la 3D-AVC, puede ser configurado para realizar la ALC sobre la base de datos sintácticos codificados en un flujo de vídeo. El codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para realizar la ALC cuando un bloque de datos de vídeo es predicho entre vistas, a partir de un bloque de una vista diferente. Cuando el bloque actual es predicho entre vistas, el codificador de vídeo 20 o el decodificador de vídeo 30 lleva a cabo la estimación de movimiento y la compensación de movimiento para determinar un vector de movimiento indicativo de un bloque de referencia correspondiente desde una vista diferente con respecto a la vista actual.

30 Para llevar a cabo la ALC, el codificador de vídeo determina en primer lugar un bloque predictivo de muestras de luma. El codificador de vídeo determina entonces bloques de muestras contiguas de luma con respecto al bloque actual de datos de vídeo que el codificador está codificando. El codificador de vídeo determina un bloque contiguo superior y un bloque contiguo izquierdo, con respecto al bloque actual, que se encuentra en la misma trama (o imagen) que el bloque actual. El codificador de vídeo también determina un bloque de referencia para el bloque actual usando un vector de movimiento de disparidad. El bloque contiguo superior de muestras de luma con respecto al bloque de referencia se indica como "UTRef_i", y el bloque contiguo izquierdo de muestras de luma con respecto al bloque de referencia se indica como "LTRef_i", donde el subíndice *i* indica un índice particular de partición o sub-partición de macro-bloque.

40 Un codificador de vídeo, configurado para realizar la ALC, también determina los bloques contiguos superior e izquierdo de muestras de luma decodificadas previamente en relación con el bloque actual, es decir, en la misma trama que el bloque actual. El bloque contiguo izquierdo de muestras de luma con respecto al bloque actual se indica como "LTDec_i," y el bloque contiguo superior de muestras de luma se indica como "UTDec_i."

45 Después de que el codificador de vídeo 20 o el decodificador de vídeo 30 determina las muestras de luma: UTDec_i, LTDec_i, UTRef_i y LTRef_i para una partición de macro-bloque en particular, el codificador de vídeo 20 o el decodificador de vídeo 30 compara los valores de las muestras de luma de UTDec_i, LTDec_i, UTRef_i y LTRef_i entre sí. Más concretamente, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden comparar las correspondientes muestras de luma de UTRef_i, LTRef_i, UTDec_i y LTDec_i entre sí de la siguiente manera:

50

$$\text{Ref_Dec}_i = 1 + \sum_{k,m=[1..4,1..psy]} \text{LTDec}_i[k,m] \times \begin{cases} 1, |\text{LTDec}_i[k,m] - \text{LTRef}_i[k,m]| < TH \\ 0, \text{de otro modo} \end{cases} + (1)$$

$$\sum_{k,m=[1..psx,1..4]} \text{UTDec}_i[k,m] \times \begin{cases} 1, |\text{UTDec}_i[k,m] - \text{UTRef}_i[k,m]| < TH \\ 0, \text{de otro modo} \end{cases},$$

$$\text{Ref_Ref}_i = 1 + \sum_{k,m=[1..4,1..psy]} \text{LTRef}_i[k,m] \times \begin{cases} 1, |\text{LTDec}_i[k,m] - \text{LTRef}_i[k,m]| < TH \\ 0, \text{de otro modo} \end{cases} +$$

$$\sum_{k,m=[1..psx,1..4]} \text{UTRef}_i[k,m] \times \begin{cases} 1, |\text{UTDec}_i[k,m] - \text{UTRef}_i[k,m]| < TH \\ 0, \text{de otro modo} \end{cases} \quad (2)$$

5 La ecuación 1 compara las muestras de luma correspondientes de LTDec_i con las muestras de luma de LTRef_i y produce una variable, "Ref_Deci", y añade la correspondiente muestra LTDec_i a Ref_Dec_i si la diferencia entre las dos muestras es menor que un valor de umbral (indicado como "TH", que es igual a 30 por omisión). La ecuación 1 también añade valor de cada muestra de UTDec_i a Ref_Dec_i si la diferencia entre UTDec_i y UTRef_i es menor que TH.

10 La ecuación 2 es similar a la ecuación 1, y añade el valor de $\text{LTRef}_i[k, m]$ a la suma Ref_Ref_i si el valor absoluto de la diferencia de LTDec_i y $\text{LTRef}_i[k, m]$ es menor que el umbral, TH. La ecuación 2 también añade el valor de $\text{UTRef}_i[k, m]$ para Ref_Ref_i si la diferencia en los valores absolutos de la diferencia entre $\text{UTDec}_i[k, m] - \text{UTRef}_i[k, m]$ es menor que TH.

15 En ambas ecuaciones, el subíndice "i" indica el índice de una partición o sub-partición de un macro-bloque en particular. Las variables "k" y "m" representan las coordenadas x e y de una muestra particular dentro de UTDec_i , UTRef_i , LTDec_i o LTRef_i . Las ecuaciones 1 y 2 se describen con fines ilustrativos. En las ecuaciones 1 y 2, UTDec_i y UTRef_i tienen una altura de 4 píxeles, y LTRef_i y LTDec_i tienen una anchura de cuatro píxeles. Como se describirá en mayor detalle a continuación, de acuerdo con las técnicas de esta divulgación, otras dimensiones para UTDec_i , LTDec_i , UTRef_i y LTRef_i son también posibles.

20 Después de calcular los valores de Ref_Ref_i y de Ref_Dec_i , el codificador de vídeo 20 o el decodificador de vídeo 30 determina una serie de parámetros de ponderación que se utilizan para ajustar los valores de luma predichos del bloque actual. Los valores de luma predichos para el bloque actual se indican como predPart_i . Para el cálculo de los parámetros de ponderación para la ALC, un codificador de vídeo determina primero un valor de una variable indicada como "W_i." W_i se determina de acuerdo con la ecuación 3, a continuación.

$$W_i = \begin{cases} 1, \text{if } (\text{Ref_Ref}_i \gg \text{CUT_TH}) == (\text{Ref_Dec}_i \gg \text{CUT_TH}) \\ ((1 \ll \text{LogWDC}) \times \text{Ref_Fef}_i + (\text{Ref_Ref}_i \gg 1)) / \text{Ref_Ref}_i, \text{de otro} \end{cases} \quad (3)$$

30 En la ecuación 3, ">>" indica un desplazamiento aritmético a la derecha, CUT_TH es un umbral con un valor igual a cuatro, y logWDC es otra variable que es igual a 15. W_i es igual a uno si el desplazamiento aritmético de $\text{Ref_Ref}_i \gg \text{CUT_TH}$ es igual al desplazamiento aritmético de $\text{Ref_Dec}_i \gg \text{CUT_TH}$. W_i es igual a $((1 \ll \text{LogWDC}) \times \text{Ref_Ref}_i + \text{Ref_Ref}_i \gg 1)$ en caso contrario.

35 Basándose en el valor calculado de W_i, el codificador de vídeo determina los valores de luma compensadas por ALC para la partición o sub-partición de macro-bloque del bloque predicho de las muestras de luma para el bloque actual. La partición compensada por ALC de los valores de luma del bloque actual se indica como $\text{predPartALC}_i[x, y]$. $\text{predPartALC}_i[x, y]$ se calcula de acuerdo con la ecuación 4 como:

$$40 \quad \text{predPartALC}_i[x,y] = \text{Min}(255, (\text{predPart}_i[x,y] \\ W_i + 2^{\text{logWDC} - 1}) \gg \text{logWDC}) \quad (4)$$

predPart_i en la ecuación 4 se calcula de acuerdo con la ecuación 5:

$$45 \quad \text{si } (\text{logWDC} \geq 1) \\ \text{predPart}_i[x, y] =$$

$$\text{Clip1} (((\text{predPartL0}_i [x, y] * w_{01} + 2^{\log WDC-1}) \gg \log WDC) + o_{01})$$

si no

$$\text{predPart}_i [x, y] = \text{Clip1} (\text{predPartL0}_i [x, y] * w_{0c} + o_{0c}) \quad (5)$$

Las técnicas de esta divulgación modifican el lenguaje del 3D-AVC 4 WD, mencionado anteriormente. Más concretamente, las técnicas de esta divulgación incluyen modificaciones para una presentación de propuesta anterior con respecto a la compensación de luminancia adaptativa que se puede encontrar en el documento de Mishurovskiy et al.: "JCT3V-B0031", ISO/IEC/JCT1/SC 29/WG 11, Shanghái, China, con fecha de oct. 2012 ("JCT3V-B0031"). Los cambios en JCT3V-B0031 se reflejan en el 3D-AVC WD 5. Los cambios que detallan algunas de las técnicas de esta divulgación con relación al documento JCT3V-B0031 se describirán ahora junto con alguna explicación. Las adiciones al borrador de trabajo se indican con las líneas que comienzan en "@". Las supresiones de la propuesta JCT3V-B0031 se indican en las líneas que comienzan con "#". El texto en **negrita** indica un encabezado de sección en el 3D-AVC 4 WD.

J.8.3.5.1 Definición de las coordenadas y tamaños de un bloque de luma a predecir

@ Sea (xM, yM) igual a la salida de la sub-cláusula 6.4.1 (la ubicación de la muestra de luma superior izquierda para el macro-bloque actual con dirección mbAddr con relación a la muestra superior izquierda de la imagen).

@ Sea (xP, yP) igual a la salida de la sub-cláusula 6.4.2.1 (la ubicación de la muestra de luma superior izquierda para la partición de macro-bloque mbPartIdx).

Sea (xB, yB) igual a la salida de la sub-cláusula 6.4.3 (la ubicación de la muestra de luma superior izquierda para el bloque de luma de tamaño 4x4 definido por Luma4x4BlkIdx que puede ser 0 ... 15).

@ Sea (xB, yB) igual a la salida de la sub-cláusula 6.4.2.2 (la ubicación de la muestra de luma superior izquierda para el bloque de luma de tamaño 4x4 con relación a la muestra superior izquierda del sub-macro-bloque).

Sea (MB8, yB8) igual a la salida del proceso 6.4.5 (la ubicación de la muestra de luma superior izquierda para el bloque de luma de tamaño 8x8 donde Luma8x8BlkIdx = (Luma4x4BlkIdx >> 2) se da como una entrada).

Las variables xCT, yCT, CBX, CBY se fijan de acuerdo con la tabla J-8-X: [Ed (MH): CBX y CBY no son nombres de variables adecuados ya que no tienen mayúsculas intermedias.]

Tabla J-8-X - Memoria descriptiva de los parámetros para obtener valores de muestra de los bloques LRec, URec, LRef, URef

#tipo_mb	indicador_tamaño_transform_8x8 igual a 1				indicador_tamaño_transform_8x8 igual a 0			
	xCT	yCT	BX	BY	xCT	yCT	BX	BY
# P_L0_16x16	xCT = xM	yCT = yM	6	6	xCT = xM + xB8	yCT = yM + yB8		
#P_L0_L0_16x8	xCT = xM + xP	yCT = yM + yP	6		xCT = xM + xB	yCT = yM + yB		
# P_L0_L0_8x16	xCT = xM + xP	yCT = yM + yP		6	xCT = xM + xB	yCT = yM + yB		
#P_8x8, # P_8x8ref0	xCT = xM + xP	yCT = yM + yP			xCT = xM + xB	yCT = yM + yB		
#P_Omitir	Indefinido				xCT = xM	yCT = yM	6	6

@ Las variables xCT, yCT, CBX, CBY se fijan de la forma siguiente:

@ Si el tipo_mb es igual a P_8x8 o P_8x8ref () y sub_mi_tipo no es P_L0_8x8,

- @ xCT se fija igual a xM + xP + xB
- @ yCT se fija igual a yM + yP + yB
- @ CBX y CBY se fijan iguales a 4.

- @ De lo contrario (el tipo_mb es igual a P_L0_16x16, P_L0_L0_16x8, o

- @ P_L0_L0_8x16, o P_8x8 y P_8x8ref () cuyos sub_mi_tipo son P_L0_8x8)

- @ xCT se fija igual a $xM + xP$
 - @ yCT se fija igual a $yM + yP$
 - @ CBX se fija igual a MbAnchuraParte (tipo_mb)
 - @ CBY se fija igual a MbAlturaParte (tipo_mb).
- 5
- @ Si una o más de las siguientes condiciones son verdaderas, W0C se fija en 1 y logWDC se fija en 15.
 - @ - xCT es menor que $(xL + 1)$ para $xL = 3$;
- 10
- @ - yCT es menor que $(yL + 1)$ para $yL = 3$;
 - @ - $(mvL0 [0] + ((xCT - xL - 1) \ll 2))$ es menor que 0, para $xL = 3$;
 - @ - $(mvL0 [1] + ((yCT - yL - 1) \ll 2))$ es menor que 0 para $yL = 3$;
- 15
- @ - $(mvL0 [0] + ((xCT + CBX) \ll 2))$ es mayor o igual que $(AnchoImagenEnMuestrasL \ll 2)$;
 - @ - $(mvL0 [1] + ((yCT + CBY) \ll 2))$ es mayor o igual que $(AlturaImagenEnMuestrasL \ll 2)$.
- 20
- @ De lo contrario, los valores de muestra de LRef, URef, LRec y URec se obtienen como se especifica en J.8.3.6.2 y J.8.3.6.3, seguido por el cálculo de las variables SumaRefcontiguo, Sumacontiguo y W0L, O0L especificadas en las sub-cláusulas J.8.3.6.4 y J.8.3.6.5 correspondientemente.
- 25
- # J.8.3.7 Proceso de inter-predicción con compensación de luminancia adaptativa para transformación de tamaño 4x4**
- @ J.8.3.7 Proceso de inter-predicción con compensación de luminancia adaptativa para tipo_mb P_8x8**
- @ Esta sub-cláusula se invoca para un macro-bloque en un componente de vista de textura solo cuando
- 30 # indicador_tamaño_transform_8x8 es igual a 0 y
- @indicador_mb_alc es igual a 1, tipo_mb es igual a P_8x8 y tanto el AnchoParteSubMb como la
- 35 AlturaParteSubMb son menos de 8.
- El proceso de inter-predicción para un macro-bloque consiste en las siguientes etapas en orden.
1. @ La sub-cláusula J.8.3.5 se invoca para obtener componentes del vector de movimiento, índices de
 - 40 referencia y ponderaciones de predicción de la siguiente manera.
 2. # El MvCnt variable se incrementa en subMvCnt.
 3. @Para cada
 - 45 # partición de macro-bloque mbPartIdx y una
 - @ partición de sub-macro-bloque subMbPartIdx, se hacen las siguientes asignaciones. [Ed (MH): Las convenciones de los nombres de variables no se siguen a continuación, se requiere reescritura.]
- # - tipo_sub_mb_Normal [mbPartIdx] = tipo_sub_mb [mbPartIdx];
- 50 # - tipo_mb_Normal = tipo_mb;
- # - Si tipomb es igual a P_L0_16x16, se aplica lo siguiente en orden.
- # - tipo_mb se fija igual a P_8x8;
- 55 # - Para mbPartIdx de 0 a NumMbPart (tipomb) - 1, ambos inclusive, #tipo_sub_mb [mbPartIdx] se fija igual a P_L0_8x8.
- # - De lo contrario, se aplica lo siguiente en orden.
- 60 # - tipo_mb se fija igual a P_8x8;
- # - Para mbPartIdx de 0 a NumMbPart (tipomb) - 1, inclusive, y tipo_sub_mb [mbPartIdx], se fija igual a P_L0_4x4.
- 65

@ Para i de 0 a NumSubMbPart - 1, tipo_sub_mb [i] se almacena en tipo_sub_mb_temp [i] y se fija igual a P_L0_4x4.

4. @ El proceso de decodificación, para muestras de inter-predicción para cada # partición de macro-bloque mbPartIdx y una @ partición de sub-macro-bloque subMbPartIdx, consiste en lo siguiente en orden.

@ - Invocar la sub-cláusula J.8.3.6 para obtener las ponderaciones de predicción en la compensación de luminancia adaptativa.

@ - Invocar la sub-cláusula 8.4.2 para obtener las muestras de inter-predicción.

@ - Invocar la sub-cláusula 8.5 para obtener los coeficientes de transformación y construir la imagen antes del proceso del filtro de desbloqueo.

5. @ Para cada # partición de macro-bloque mbPartIdx y una @ partición de sub-macro-bloque subMbPartIdx, se hacen las siguientes asignaciones.

- tipo_mb = tipo_mb_Normal
- tipo_sub_mb [mbPartIdx] = tipo_sub_mb_Normal [mbPartIdx]
@tipo_sub_mb [i] se restaura a tipo_sub_mb_temp [i].

Estas revisiones al Anexo J, secciones sub-cláusulas J.8.3.5.1 a J.8.3.7 del 3D-AVC WD4, modifican las técnicas de compensación de luminancia adaptativa descritas en la misma. Las técnicas de compensación de luminancia adaptativa pueden ser realizadas por un codificador de vídeo, tal como el codificador de vídeo 20, o el decodificador de vídeo 30. Las modificaciones pueden dar como resultado la reducción de la complejidad de cálculo en relación con las técnicas descritas en la propuesta V-B0031 JCT3.

Como se ha descrito brevemente más arriba, las técnicas de la propuesta V-B0031 JCT3 pueden hacer que un codificador de vídeo compatible lleve a cabo la compensación de movimiento para determinar los valores de ponderación de la ALC, indicados como W_i , para cada (sub-)partición de macro-bloque, para determinar muestras de luma contiguas entre vistas para la compensación de luminancia adaptativa. Si se divide un macro-bloque, por ejemplo, en sub-particiones, la complejidad de cálculo de la realización de la compensación de movimiento y el cálculo de W_i para cada bloque de tamaño 4x4 puede ser alta.

De acuerdo con un ejemplo de las técnicas de esta divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar un conjunto de ponderaciones de predicción solo una vez por partición de un macro-bloque de datos de vídeo, o incluso una vez por macro-bloque (por ejemplo, para una partición particular del macro-bloque, y reutilizados para cada partición adicional del macro-bloque). Más en particular, los cambios anteriores al 3D-AVC WD4 modifican el proceso de determinación de la ubicación de los bloques de muestras de luma a partir de la imagen de referencia.

Una vez más, con respecto a los cambios anteriores del documento WD 4, si el macro-bloque actual es una partición de macro-bloque de tamaño 8x8 que no tiene un tipo de sub-macro-bloque de tamaño 8x8, la coordenada x que indica la posición de los bloques contiguos izquierdo y superior de muestras de luma (x_{CT}) es igual a la coordenada x superior izquierda de la muestra de luma superior izquierda para el macro-bloque actual, más la coordenada x de la muestra de luma superior izquierda de la partición del macro-bloque actual (x_P), más la coordenada x de la muestra de luma superior izquierda de la sub-partición del macro-bloque (x_B). La coordenada y , indicativa de la posición de los bloques de muestras contiguas superiores e izquierdas de luma (y_{CT}), es igual a la coordenada y de la muestra de luma superior izquierda del macro-bloque actual (y_M), más la coordenada y de la muestra de luma superior izquierda de la partición del macro-bloque (y_P), más la coordenada y de la muestra de luma superior izquierda del bloque de luma de tamaño 4x4, con relación a la muestra superior izquierda del sub-macro-bloque (y_B). En este caso, la altura y la anchura de los bloques de muestras de luma es igual a las de las muestras de luma de tamaño 4x4.

De lo contrario, cuando el indicador_tamaño_transform_8x8 es igual a 1, x_{CT} es igual a $x_M + x_P$, y_{CT} es igual a $y_M + y_P$, y las alturas de los bloques de muestras contiguas izquierdas de luma (LTR_{ref} y LTD_{dec}) son iguales a la altura de la partición del macro-bloque actual. Las anchuras de los bloques de muestras contiguas superiores de luma (UTR_{ref} y UTD_{dec}) son iguales a la anchura de la partición del macro-bloque actual. Por lo tanto, el mismo conjunto de ponderaciones de predicción se determina para cada partición de macro-bloque, en lugar de individualmente para cada partición de sub-macro-bloque, independientemente del tamaño de transformación del macro-bloque.

En algunos otros ejemplos de las técnicas de esta divulgación, incluso para las particiones de macro-bloque de tamaño 8x8, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar un conjunto de ponderaciones de predicción, W_i , independientemente de si los vectores de movimiento para cada uno de los

bloques de luma de tamaño 4x4 de la partición del macro-bloque son los mismos o no. En este ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 obtiene las ponderaciones de predicción, W_i , en base a ponderaciones de ALC determinadas para un bloque de tamaño 4x4 de muestras de luma de la partición del macro-bloque. En algunos ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar las técnicas de este ejemplo cuando el bloque actual tiene un tamaño de transformación más grande que las muestras de luma de tamaño 4x4. El bloque de tamaño 4x4 que el codificador de vídeo utiliza para obtener ponderaciones de predicción puede ser cualquier bloque de luma de tamaño 4x4 dentro de la partición de MB de tamaño 8x8 en algunos casos.

En algunos otros ejemplos, cuando se realiza la ALC, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar el bloque de tamaño 4x4 de las muestras de luma que tiene el valor absoluto máximo del vector de movimiento (es decir, la suma del valor absoluto del componente horizontal y del valor absoluto del componente vertical del vector de movimiento).

En otros ejemplos más, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden obtener y utilizar un conjunto de ponderaciones de predicción para todo el macro-bloque actual. El codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar las ponderaciones de la ALC para el macro-bloque actual utilizando el conjunto de ponderaciones de predicción obtenidas para la partición de macro-bloque que contiene la muestra de luma superior izquierda del macro-bloque. Si la partición de MB contiene varios bloques de luma que tienen diferentes vectores de movimiento, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar las ponderaciones de predicción, W_i , para el macro-bloque en base a las ponderaciones obtenidas para el bloque superior izquierdo de la partición del macro-bloque.

En algunos ejemplos, de conformidad con las técnicas de esta descripción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar diferentes tamaños de $LTRef_i$, $LTDc_i$, $UTDec_i$ y $UTRef_i$. Por ejemplo, cuando se admite la compensación de iluminación para un bloque dado que tiene un tamaño de $N \times M$, (donde N es la anchura y la M es la altura, y pueden ser uno entre 4, 8 o 16), en lugar de utilizar $N \times 4$ filas superiores y $4 \times M$ columnas izquierdas, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar menos filas o menos columnas para la obtención de las ponderaciones de predicción para la ALC.

En otro ejemplo, si N (o M) es igual a 16, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar las filas superiores (o columnas izquierdas) de $UTRef_i$, $UTDec_i$, $LTRef_i$ y $LTDc_i$ de modo tal que los bloques de muestras de luma elegidos tengan un tamaño de 8×4 (o 4×8). En algunos otros ejemplos de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede también ser configurado para no invocar la compensación de movimiento, sino más bien para tomar un vector de movimiento existente, redondear el vector y utilizar el vector de movimiento redondeado para determinar las ubicaciones de los bloques de luma de la imagen de referencia (es decir, $UTRef_i$ y $LTRef_i$).

Por ejemplo, cuando se obtienen las muestras de regiones superiores e izquierdas en la imagen de referencia (que es una imagen de referencia entre vistas), en vez de invocar la compensación de movimiento para obtener las muestras de luma en base al mismo vector de movimiento que el bloque actual, el codificador de vídeo 20 o el decodificador de vídeo 30 puede redondear un vector de movimiento. Por ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede tomar un vector de movimiento existente (por ejemplo, $mv[0]$, $mv[1]$), y redondear el vector de movimiento a un número entero de pel. Para redondear el vector de movimiento a un número entero de pel, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar una de las ecuaciones 6 a 8, más adelante (cuando el vector de movimiento ya es un vector de movimiento de un número entero de pel, no es necesaria la interpolación):

$$mv[0] = (mv[0] \gg 2) \ll 2 \text{ y } mv[1] = (mv[1] \gg 2) \ll 2, \text{ o} \quad (6)$$

$$mv[0] = ((mv[0] + 2) \gg 2) \ll 2, \text{ y } mv[1] = ((mv[1] + 2) \gg 2) \ll 2 \text{ o} \quad (7)$$

$$mv[0] = ((mv[0] + \text{signo}(mv[0]) * 2) \gg 2) \ll 2, \text{ y}$$

$$mv[1] = ((mv[1] + \text{signo}(mv[1]) * 2) \gg 2) \ll 2. \quad (8)$$

Como alternativa, el codificador de vídeo 20 o el decodificador de vídeo 30 puede redondear el vector de movimiento hasta la mitad de pel, por ejemplo, usando la ecuación 9, a continuación:

$$mv[0] = (mv[0] \gg 1) \ll 1 \text{ y } mv[1] = (mv[1] \gg 1) \ll 1. \quad (9)$$

El codificador de vídeo 20 o el decodificador de vídeo 30 puede también convertir el vector de movimiento para obtener los bloques de muestras de luma superior e izquierda a partir de (un diferente) vector de movimiento utilizado para compensar el movimiento del bloque actual.

En otro ejemplo, en vez de realizar la compensación de movimiento para un bloque de luma contiguo superior o izquierdo que tenga un tamaño igual a 4×4 o mayor que 4×4 , el codificador de vídeo 20 o el decodificador de vídeo

30 puede considerar solo las muestras de luma en una fila superior y una columna izquierda al obtener las ponderaciones de predicción, W_i . En otro ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar solamente la fila superior y la columna izquierda, pero puede obtener la ubicación de los bloques de referencia de las muestras de luma utilizando un vector de movimiento que se haya redondeado a un número entero de pel.

En algunos otros ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar solo un subconjunto de muestras de luma de un bloque de muestras de luma contiguas al obtener ponderaciones de predicción. Por ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede usar ambos extremos de una fila o columna para obtener las ponderaciones de predicción. En otro ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar muestras de luma en ambos extremos del bloque de muestra de luma, así como la(s) muestra(s) media(s) de la fila o columna al determinar las ponderaciones para la ALC.

En otro ejemplo más, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar hasta cuatro muestras de luma de una fila o columna al determinar las ponderaciones de predicción para la ALC. En este ejemplo, si una fila / columna de muestras de luma contiguas (en relación con el bloque actual) tiene una longitud de N muestras (en la que $N = 4 \cdot 2^S$, $S=0, 1$ ó 2), la posición horizontal / vertical $x[i]$ de las muestras elegidas, con respecto a la coordenada horizontal o vertical de la muestra superior izquierda del bloque, puede calcularse como: $x[i]=i \ll S$, donde i está en la gama entre 0 y 3 , inclusive.

En otra alternativa más, el codificador de vídeo 20 o el decodificador de vídeo 30 puede reducir el muestreo de las muestras de luma de una fila o columna, el codificador de vídeo 20 o el decodificador de vídeo 30 también puede filtrar y reducir el muestreo de la luma de una fila o columna al determinar las ponderaciones de la ALC.

Como alternativa, cuando se admite la compensación de iluminación para un bloque dado que tiene un tamaño de $N \times M$, en el que N es la anchura y la M es la altura y al menos uno de ellos es igual a 16, en lugar de utilizar las $N \times 4$ filas superiores y las $4 \times M$ columnas izquierdas, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar menos filas o menos columnas para realizar la obtención de las ponderaciones de predicción para la ALC.

En otros ejemplos más, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar muestras en el interior de la región de muestras de luma contiguas superiores de tamaño 16×4 (o la región izquierda de muestras de luma de tamaño 4×16) de forma que las muestras de luma formen una región de tamaño 8×4 (o 4×8) o 4×4 . El codificador de vídeo 20 o el decodificador de vídeo 30 también puede utilizar muestras filtradas de la región superior de tamaño 16×4 (o la región izquierda de tamaño 4×16) que pueden formar una región de tamaño 8×4 (o 4×8) o 4×4 . También debería entenderse que cualquiera de las técnicas de esta divulgación se puede realizar independientemente de las dimensiones, N o M , del macro-bloque.

En varios ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar las dimensiones de las regiones de muestras contiguas izquierdas y contiguas superiores, independientemente del tamaño de transformación del macro-bloque actual. El codificador de vídeo 20 o el decodificador de vídeo 30 puede también seleccionar las dimensiones de las regiones de muestras contiguas izquierdas y superiores, independientemente del tamaño de transformación del bloque actual, pero en base al tamaño de la partición o sub-partición del macro-bloque actual.

El codificador de vídeo 20 o el decodificador de vídeo 30 también puede realizar la identificación de muestras de referencia independientes de la partición de MB: en este procedimiento, un MB siempre es filtrado por un conjunto de ponderaciones generadas independientemente de la partición de MB.

En diversos ejemplos, como alternativa, el codificador de vídeo 20 o el decodificador de vídeo 30 puede fijar las muestras de referencia de luma superiores e izquierdas para la ALC en los bloques de luma superiores de tamaño 4×4 e izquierdos de tamaño 4×4 , del bloque superior izquierdo de tamaño 4×4 del macro-bloque actual. Además, el codificador de vídeo 20 o el decodificador de vídeo 30 puede fijar el vector de movimiento de ese bloque, que se utiliza para obtener el predictor de las muestras de referencia, en una vista de referencia.

En algunos ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede fijar las muestras de referencia superior e izquierda en los bloques de luma superiores de tamaño 4×4 e izquierdos de tamaño 4×4 del bloque de superior izquierdo de tamaño 4×4 del macro-bloque actual. El codificador de vídeo 20 o el decodificador de vídeo 30 también puede determinar el vector de movimiento utilizado para determinar el bloque predictor de muestras de referencia de luma en la vista de referencia, mediante el uso de uno o más vectores de movimiento de los bloques del macro-bloque actual.

En algunas otras técnicas de esta divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para realizar la identificación de muestras de referencia, dependiente de la partición del macro-bloque, pero independiente de la partición de sub-bloques. En estas técnicas, el codificador de vídeo 20 o el decodificador de vídeo 30 puede obtener un conjunto de ponderaciones de la ALC para cada partición de macro-bloque. El codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para filtrar conjuntamente píxeles

dentro de una partición de macro-bloque utilizando el conjunto de ponderaciones, independientemente de la partición del sub-bloque. Por ejemplo, para una partición de MB de tamaño 8x8, si la partición de tamaño 8x8 se divide además en cuatro bloques de tamaño 4x4, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden utilizar el mismo conjunto de coeficientes de filtro para los cuatro bloques, y pueden filtrar los bloques conjuntamente.

En otro ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede fijar las muestras de referencia superior e izquierda en el bloque superior de tamaño 4x4 y en el bloque superior izquierdo de tamaño 4x4 de una partición del macro-bloque. En este ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 utiliza el vector de movimiento del bloque superior de tamaño 4x4 y del bloque izquierdo de tamaño 4x4 para determinar la ubicación del bloque predictivo de las muestras de luma a partir de la vista de referencia.

En otra alternativa, si una partición de macro-bloque tiene un tamaño de MxN, y M o N es mayor que 8, o M o N tiene un tamaño igual a 8, pero la partición del macro-bloque tiene una partición de sub-bloque que es de tamaño 8x8, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar la región de muestras de referencia luma izquierda de tal manera que la región tenga un tamaño de 4xN y una región de las principales muestras de referencia, de tal manera que la región de muestras de referencia superior tenga un tamaño de Mx4. El codificador de vídeo 20 o el decodificador de vídeo 30 también puede seleccionar la región de muestras de referencia izquierda de tal manera que la región izquierda de muestras de referencia de luma tenga un tamaño de 4x8 y la región derecha de muestras de referencia de tal manera que la región derecha de muestras de referencia de luma tenga un tamaño de 8x4. El codificador de vídeo 20 o el decodificador de vídeo 30, tales que las regiones de las muestras de referencia de luma izquierda y superior tengan un tamaño de 4xN/2 y M/2x4, respectivamente.

En varios ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden realizar la identificación de muestras de referencia, dependientes de la partición del macro-bloque y dependientes de la partición del sub-bloque. En este ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 obtiene un conjunto de ponderaciones de predicción de la ALC para cada partición de MB. Sin embargo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar diferentes técnicas para identificar las muestras de referencia para cada partición de sub-bloque. Si la partición del macro-bloque es una partición de tamaño 8x8, y la partición de sub-bloque tiene un tamaño de: dos bloques de tamaño 8x4 (mxn) o dos bloques de tamaño 4x8 (mxn), el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar la región izquierda de muestras de referencia de modo que la región de muestras de referencia contiguas izquierdas tenga un tamaño de 4xn y la de muestras de luma de referencia superiores tenga un tamaño de mx4.

En una alternativa, si la partición del macro-bloque es de tamaño 8x8 y la partición de sub-bloque es de dos bloques de tamaño 8x4 (mxn) o dos bloques de tamaño 4x8 (mxn), el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar el vector de movimiento indicativo de la ubicación del bloque predictivo de las muestras de luma para el bloque actual, sobre la base de uno de los vectores de movimiento indicativos de las ubicaciones de los bloques contiguos superiores o izquierdos de las muestras de luma de referencia. En otra alternativa, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar los vectores de movimiento que indican la ubicación de los bloques de muestras de referencia contiguas superiores y/o izquierdas al determinar el vector de movimiento indicativo del bloque predictivo de las muestras de luma para el bloque actual.

Las técnicas descritas en el presente documento para modificaciones en el borrador de trabajo de 3D-AVC incluyen técnicas que hacen que un codificador, configurado de acuerdo con las técnicas de esta divulgación, lleve a cabo la compensación de movimiento para determinar las ponderaciones para la ALC solo una vez por partición de macro-bloque. El codificador de vídeo utiliza luego el mismo vector de movimiento para determinar las muestras de luma de referencia para cada partición de macro-bloque y sub-particiones cualesquiera dentro de la partición del macro-bloque.

Por lo tanto, en un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación para la partición del bloque de vídeo, calcule un bloque predicho para la partición del bloque de vídeo utilizando las ponderaciones de predicción y utilizando la compensación de la iluminación, y codifique (por ejemplo, codifique o decodifique) la partición del bloque de vídeo utilizando el bloque predicho.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo a modo de ejemplo, que puede estar configurado para realizar la compensación de luminancia adaptativa, de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 recibe datos de vídeo a codificar. En el ejemplo de la FIG. 2, el decodificador de vídeo 20 incluye una unidad de selección de modalidad 40, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54, una unidad de codificación por entropía 56 y una memoria de imágenes de referencia 64. A su vez, la unidad de selección de modalidad 40 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44, una unidad de intra-predicción 46 y una unidad de

partición 48.

Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantización inversa 58, una unidad de procesamiento de transformación inversa 60 y un 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 de bucle adicionales (en bucle o tras el 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 (tal como un filtro en bucle).

La unidad de selección de modalidad 40 puede recibir datos de vídeo en bruto en forma de bloques de una o más vistas. En la norma 3D-AVC, las vistas pueden corresponder a una vista del ojo izquierdo y a una vista del ojo derecho. La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter (por ejemplo, la inter-predicción intra-vistas, también denominada predicción temporal, o la predicción entre vistas), por ejemplo, en base a los resultados de error, y proporciona el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloque residual, y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia. En algunos ejemplos, la unidad de selección de modalidad 40 puede determinar que un bloque debería predecirse entre vistas, en base a los datos de vídeo desde una vista diferente. La unidad de selección de modalidad 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de intra-modalidad, información de partición y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con 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 los bloques de vídeo. De manera similar, la unidad de estimación de movimiento 42 puede realizar la estimación de disparidad para la predicción entre vistas, lo cual da como resultado un vector de movimiento de disparidad. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque de un bloque de vídeo dentro de una imagen actual, con respecto a un bloque predictivo dentro de una imagen de referencia (u otra unidad codificada), con respecto al bloque actual que está siendo codificado dentro de la imagen actual (u otra unidad codificada).

En diversos ejemplos de acuerdo con las técnicas de esta divulgación, la unidad de estimación de movimiento 42 puede determinar un vector de movimiento que indica el desplazamiento del bloque actual en relación con un bloque predictivo a partir de una vista diferente, por ejemplo, cuando el bloque actual es predicho entre vistas. La unidad de estimación de movimiento 42 también puede estar configurada para determinar los vectores de movimiento que indican la(s) ubicación(es) de uno o más bloques o regiones de las muestras de luma de referencia que el codificador de vídeo 20 utiliza para determinar las ponderaciones al realizar la ALC. La unidad de estimación de movimiento 42 puede calcular los vectores de movimiento (por ejemplo, vectores de movimiento temporal y/o de disparidad) que tienen una precisión de imagen fraccionada, por ejemplo, la mitad o un cuarto de la precisión de píxeles. De acuerdo con ciertas técnicas de esta divulgación, la unidad de compensación de movimiento 44 puede estar configurada para realizar el redondeo de los vectores de movimiento cuando se realiza la ALC, por ejemplo, el redondeo de un vector de movimiento fraccional en pel hasta un vector de movimiento de precisión de un número entero de pel o de medio-pel.

Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque a codificar, en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de fracciones de píxel 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 fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y a las posiciones de fracciones de píxel, y emitir un vector de movimiento con una precisión de fracciones de píxel.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento inter-codificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La unidad de estimación/ disparidad de movimiento 42 también puede configurarse para realizar la predicción entre vistas, en cuyo caso la unidad de estimación/ disparidad de movimiento 42 puede calcular vectores de desplazamiento entre los bloques de una imagen de vista (por ejemplo, la vista 0) y los bloques correspondientes de una imagen de vista de referencia (por ejemplo, la vista 1). En general, los datos para un vector de movimiento pueden incluir una lista de imágenes de referencia, un índice en la lista de imágenes de referencia (ref_idx), un componente horizontal y un componente vertical. La imagen de referencia puede seleccionarse a partir de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Para cada lista (es decir, la Lista 0 o la Lista 1), el codificador de vídeo 20 puede seleccionar las

entradas en base al orden ascendente del índice de referencia.

5 La unidad de estimación de movimiento 42 puede generar y enviar un vector de movimiento que identifica el bloque predictivo de la imagen de referencia a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44. Es decir, la unidad de estimación de movimiento 42 puede generar y enviar datos de vectores de movimiento que identifican la lista de imágenes de referencia que contiene el bloque predictivo, un índice en la lista de imágenes de referencia que identifica la imagen del bloque predictivo y un componente horizontal y vertical para localizar el bloque predictivo dentro de la imagen identificada.

10 La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para una partición del macro-bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo
15 al que apunta el vector de movimiento en una de las listas de imágenes de referencia.

La unidad de compensación de movimiento 44 puede también realizar ciertas técnicas relacionadas con la ALC, de acuerdo con las técnicas de esta divulgación. En algunos ejemplos, la unidad de compensación de movimiento 44 puede estar configurada para determinar bloques de muestras de luma contiguas superiores e izquierdas con respecto al bloque actual. Los bloques de muestras contiguas superiores e izquierdas pueden comprender regiones de muestras de luma de los bloques previamente codificados de la misma trama que el bloque actual (por ejemplo, las regiones UTDec y LTDec descritas anteriormente), así como las regiones de muestras de luma que son contiguas del bloque de referencia (por ejemplo, LTRef y UTDec, descritas anteriormente).

25 La unidad de compensación de movimiento 44 puede determinar la ubicación de las regiones de las muestras, así como la ubicación de los valores de las muestras predictivas de luma para el bloque actual, sobre la base de uno o más vectores de movimiento generados por la unidad de estimación de movimiento 42. Si el bloque actual tiene la ALC habilitada, una vez que la unidad de compensación de movimiento 44 determina las regiones de muestras de luma contiguas superior e izquierda para el bloque actual, la unidad de compensación de movimiento 44 compara los valores absolutos de las muestras correspondientes de los bloques para generar las variables Ref_Ref y Ref_Dec, y el factor de ponderación para el bloque actual, W_i , como se ha descrito anteriormente. Una vez que la unidad de compensación de movimiento 44 determina el factor de ponderación, W_i para la partición o sub-partición del macro-bloque, la unidad de compensación de movimiento 44 puede ajustar el bloque predictivo de los valores de luma para el bloque actual, $predPart_i$, en el factor de ponderación W_i de conformidad con la ecuación 4, descrita anteriormente.

35 De acuerdo con las técnicas de esta divulgación, la unidad de compensación de movimiento 44 puede estar configurada para obtener ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición bloque de vídeo, una vez para la partición del bloque de vídeo, de tal manera que la partición del bloque de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente del tamaño de transformación para la partición del bloque de vídeo. La unidad de compensación de movimiento 44 puede estar configurada para realizar la compensación de la iluminación de las muestras de luma, adicionalmente de acuerdo con las técnicas de esta divulgación. Por ejemplo, cuando un macro-bloque actual es predicho unidireccionalmente a partir de una imagen de referencia entre vistas en un fragmento P, la unidad de compensación de movimiento 44 puede realizar la compensación de la iluminación para las muestras de luma del macro-bloque actual, de acuerdo con las técnicas de esta divulgación. Del mismo modo, la unidad de compensación de movimiento 44 puede aplicar las ponderaciones de predicción a un bloque de referencia para una partición de un macro-bloque actual, para calcular un bloque predictivo para la partición.

50 El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que está siendo codificado, generando valores de diferencia de píxel, como se analiza posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en base a los componentes de luma, tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por parte del decodificador de vídeo 30 a la hora de decodificar los bloques de vídeo del fragmento de vídeo.

60 La unidad de intra-predicción 46 puede intra-predicir un bloque actual, como alternativa a la inter-predicción llevada a cabo 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 intra-predicción 46 puede determinar una modalidad de intra-predicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intra-predicción 46 puede codificar un bloque actual usando varias modalidades de intra-predicción, por ejemplo, durante diferentes pasadas de codificación, y la unidad de intra-predicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad adecuada de intra-predicción a usar, entre las modalidades probadas.

65

- 5 Por ejemplo, la unidad de intra-predicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intra-predicción probadas, y seleccionar la modalidad de intra-predicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina en general una magnitud de distorsión (o errores) entre un bloque codificado y un bloque original, no codificado, que se codificó para generar el bloque codificado, así como una velocidad binaria (es decir, un número de bits) usada para generar el bloque codificado. La unidad de intra-predicción 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modalidad de intra-predicción presenta el mejor valor de velocidad-distorsión para el bloque.
- 10 Después de seleccionar una modalidad de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información, indicativa de la modalidad de intra-predicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica la modalidad de intra-predicción seleccionada. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de intra-predicción y una pluralidad de tablas de índices de modalidades de intra-predicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de la modalidad de intra-predicción más probable, una tabla de índices de modalidades de intra-predicción y una tabla modificada de índices de modalidades de intra-predicción, a usar en cada uno de los contextos.
- 15 El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 del bloque de vídeo original que está siendo codificado. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformaciones 52 aplica una transformación, tal como una transformación discreta del coseno (DCT) o una transformación conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformaciones 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También podrían usarse transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformaciones 52 aplica la transformación al bloque residual, generando un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual, desde un dominio de valores de píxel a un dominio de transformaciones, tal como un dominio de frecuencia.
- 20 La unidad de procesamiento de transformaciones 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o a todos, los coeficientes. El grado de cuantización puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar a continuación un escaneo de la matriz, incluyendo los coeficientes de transformación cuantizados. Como alternativa, la unidad de codificación por entropía 56 puede realizar el escaneo.
- 25 Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto y basada en sintaxis (SBAC), la codificación por entropía mediante la partición en intervalos de probabilidades (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto puede basarse en bloques contiguos. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.
- 30 La unidad de cuantización inversa 58 y la unidad de transformación inversa 60 aplican la cuantización inversa y la transformación 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 una de las imágenes 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 fracciones de píxel para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una imagen posterior.
- 35 El codificador de vídeo 20 puede generar una serie de elementos sintácticos, como se ha descrito anteriormente, que pueden ser codificados por la unidad de codificación por entropía 56 u otra unidad de codificación del codificador de vídeo 20. En algunos ejemplos, el codificador de vídeo 20 puede generar y codificar elementos sintácticos para un flujo de bits de 3D-AVC, como se ha descrito anteriormente, donde de nuevo este flujo de bits de
- 40
- 45
- 50
- 55
- 60
- 65

3DVC puede ser retro-compatible con la norma H.264/AVC.

De esta manera, el codificador de vídeo 20 puede estar configurado para obtener ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación para la partición del bloque de vídeo, calcular un bloque predicho para la partición del bloque de vídeo utilizando las ponderaciones de predicción y usando la compensación de la iluminación, y codificar la partición del bloque de vídeo usando el bloque predicho.

La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo 30 a modo de ejemplo que puede configurarse para realizar la compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una unidad de predicción 81 con una unidad de compensación de movimiento 72 y una unidad de intra-predicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, un sumador 80 y una memoria de imágenes de referencia 82.

Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos relacionados con la predicción a la unidad de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos al nivel del fragmento de vídeo y/o al nivel del bloque de vídeo.

Por ejemplo, el decodificador de vídeo 30 puede recibir un cierto número de unidades de NAL que tienen una cabecera de unidad de NAL que identifica un tipo de datos almacenados en la unidad de NAL (por ejemplo, datos de VCL y datos no de VCL). Los conjuntos de parámetros pueden contener la información de cabecera a nivel de secuencia, tal como un SPS, PPS u otro conjunto de parámetros descrito anteriormente.

Cuando un fragmento de vídeo se codifica como un fragmento intra-codificado (I), el módulo de intra-predicción 74 del módulo de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, en base a una modalidad de intra-predicción señalizada, y datos de bloques previamente decodificados de la imagen actual, a la vista de los conjuntos de parámetros activos. Cuando la imagen es codificada como un fragmento inter-codificado (es decir, B, P o PB generalizado (GPB)), la unidad de compensación de movimiento 72 de la unidad de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, en base a los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 80, nuevamente a la vista de los conjuntos de parámetros activos. Los bloques predictivos pueden ser generados a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia, la Lista 0 y la Lista 1 (o una lista combinada, la Lista c), usando técnicas de construcción por omisión, en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intra-predicción o inter-predicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter-predicción para cada bloque de vídeo inter-codificado del fragmento y otra información, para decodificar los bloques de vídeo en el fragmento de vídeo actual.

La unidad de decodificación por entropía o la unidad de compensación de movimiento 72 también puede determinar los vectores de movimiento que se utilizan al realizar la ALC en un bloque predicho entre vistas. La unidad de compensación de movimiento 72 puede determinar la ubicación de un bloque predictivo entre vistas de las muestras de luma que se utilizan para predecir los valores de luma para el bloque actual (predPart) en base a un vector de movimiento decodificado usando la unidad de decodificación por entropía 70. La unidad de decodificación por entropía 70 y la unidad de compensación de movimiento 72 también pueden determinar uno o más vectores de movimiento que indican la posición de las regiones de bloques de muestras de luma contiguas superiores e izquierdas, LTRef_i y UTRef_i a partir del mismo bloque de referencia como predPart_i, el bloque de luma predictivo a partir de la vista de referencia.

La unidad de compensación de movimiento 72 puede también determinar las regiones de bloques de muestras de luma contiguas superiores e izquierdas con respecto al bloque actual, LTDec_i y UTDec_i. LTDec_i y UTDec_i se

encuentran en la misma trama que el bloque actual. La unidad de compensación de movimiento 72 puede realizar la ALC en el bloque actual mediante la determinación de la diferencia entre las muestras correspondientes de $UTRef_i$ y $UTDec_i$ y $LTRef_i$ y $LTDec_i$, como se ha descrito anteriormente en la ecuación para generar las variables Ref_Ref_i y Ref_Dec_i para la partición de o sub-partición del macro-bloque actual. La unidad de compensación de movimiento 72 puede determinar el factor de ponderación W_i para la realización de la ALC en la partición o sub-partición del macro-bloque actual, etc., de acuerdo con la ecuación 3, más arriba. La unidad de compensación de movimiento 72 ajusta los valores de luma predichos de $predPart_i$ en W_i , de acuerdo con la ecuación 4, más arriba. La salida de la ponderación de $predPart_i$ es el bloque predictivo ponderado de muestras de luma, $predPartALC_i$.

De esta manera, la unidad de compensación de movimiento 72 puede estar configurada para realizar la compensación de la iluminación de acuerdo con las técnicas de esta divulgación, por ejemplo, para obtener ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición de bloque de vídeo, una vez para la partición del bloque de vídeo, de tal manera que la partición del bloque de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación para la partición de bloques de vídeo, y para calcular un bloque predicho para la partición del bloque de vídeo utilizando las ponderaciones de predicción y usando la compensación de la iluminación.

La unidad de cuantización inversa 76 cuantiza de manera inversa, es decir, des-cuantiza, los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse.

La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación con el fin de generar bloques residuales en el dominio de píxeles. El procesamiento de transformación inversa 78 puede determinar la manera en la que las transformaciones se aplicaron a los datos residuales.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual, en base a los vectores de movimiento y a otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de transformación inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados, con el fin de eliminar distorsiones de efecto pixelado. Otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) también pueden ser usados para allanar las transiciones de píxeles, o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una imagen dada son a continuación almacenados en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

De esta manera, el decodificador de vídeo 30 de la FIG. 3 puede estar configurado para obtener las ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición del bloque de vídeo, una vez para la partición del bloque de vídeo, de tal manera que la partición del bloque de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación de la partición de bloques de vídeo, para calcular un bloque predicho para la partición del bloque de vídeo utilizando las ponderaciones de predicción y usando la compensación de la iluminación, y decodificar la partición del bloque de vídeo usando el bloque predicho.

La FIG. 4 es un diagrama conceptual que ilustra un patrón de predicción de la norma 3D-AVC a modo de ejemplo. En el ejemplo de la FIG. 4, se ilustran ocho vistas y se ilustran doce ubicaciones temporales para cada vista. En general, cada fila en la FIG. 4 corresponde a una vista, mientras que cada columna indica una ubicación temporal. Cada una de las vistas se puede identificar usando un identificador de vista ("id_vista"), que puede ser utilizado para indicar una ubicación de cámara relativa con respecto a las otras vistas. En el ejemplo mostrado en la FIG. 4, los identificadores de vista se indican con "S0" a "S7", aunque también se pueden utilizar identificadores numéricos de vista. Además, cada una de las ubicaciones temporales se puede identificar usando un valor de recuento de orden de imágenes (POC), que indica una orden de visualización de las imágenes. En el ejemplo mostrado en la FIG. 4, los valores de POC se indican con "T0" a "T11".

Las imágenes en la FIG. 4 se indican usando un bloque sombreado que incluye una letra, que designa si la imagen correspondiente está intra-codificada (es decir, una trama I), o inter-codificada en una dirección (es decir, como una trama P) o en múltiples direcciones (es decir, como una trama B). En general, las predicciones se indican mediante flechas, donde la imagen a la que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. Por ejemplo, la trama P de la vista S2 en la ubicación temporal T0 se predice a partir de la trama I de la vista S0 en

la ubicación temporal T0. Cada una de las imágenes mostradas en la FIG. 4 puede denominarse un componente de vista.

Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de la norma 3D-AVC pueden codificarse predictivamente con respecto a las imágenes en diferentes ubicaciones temporales. Por ejemplo, la trama B de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la trama I de la vista S0 en la ubicación temporal T0, lo cual indica que la trama B se predice a partir de la trama I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo de múltiples vistas, las imágenes pueden ser predichas entre vistas. Es decir, un componente de vista puede utilizar los componentes de vista en otras vistas para referencia. En la norma 3D-AVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista en otra vista es una referencia de inter-predicción. Las referencias potenciales entre vistas pueden ser señalizadas en la extensión de la norma 3D-AVC del SPS y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, lo cual habilita el ordenamiento flexible de las referencias de inter-predicción o de predicción entre vistas.

La FIG. 4 proporciona varios ejemplos de la predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la FIG. 4, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como predichas entre vistas a partir de imágenes de vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la trama B de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las tramas B de la vista S1 en las ubicaciones temporales T0 y T2, así como las tramas B de las vistas S0 y S2 en la ubicación temporal T1.

En el ejemplo de la FIG. 4, la letra "B" mayúscula y la "b" minúscula están concebidas para indicar diferentes relaciones jerárquicas entre las imágenes, en lugar de diferentes metodologías de codificación. En general, las tramas con "B" mayúscula están relativamente más altas en la jerarquía de predicción que las tramas con "b" minúscula. La FIG. 4 también ilustra las variaciones en la jerarquía de predicción utilizando diferentes niveles de sombreado, donde las imágenes con una mayor magnitud de sombreado (es decir, relativamente más oscuras) están más altas en la jerarquía de predicción que aquellas imágenes que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todas las tramas I en la FIG. 4 se ilustran con sombreado completo, mientras que las tramas P tienen un sombreado algo más claro, y las tramas B (y las tramas con b minúscula) tienen diversos niveles de sombreado entre sí, pero siempre más claros que el sombreado de las tramas P y las tramas I.

En general, la jerarquía de predicción se relaciona con índices del orden de vistas, en cuanto a que las imágenes relativamente más altas en la jerarquía de predicción deberían ser decodificadas antes de la decodificación de imágenes que están relativamente más bajas en la jerarquía, de tal modo que esas imágenes relativamente más altas en la jerarquía se puedan utilizar como imágenes de referencia durante la decodificación de las imágenes relativamente más bajas en la jerarquía. Un índice de orden de vistas es un índice que indica el orden de decodificación de componentes de vista en una unidad de acceso. Los índices de orden de vistas pueden estar implícitos en un conjunto de parámetros, tal como un SPS. De esta manera, las imágenes utilizadas como imágenes de referencia pueden ser decodificadas antes de la decodificación de las imágenes que se codifican con referencia a las imágenes de referencia.

En algunos casos, un subconjunto de todo un flujo de bits puede ser extraído para formar un sub-flujo de bits que aún se ajusta a la norma 3D-AVC. Hay muchos posibles sub-flujos de bits que las aplicaciones específicas pueden requerir, en base a, por ejemplo, un servicio proporcionado por un servidor, la capacidad, el apoyo y las capacidades de los decodificadores de uno o más clientes y/o la preferencia de uno o varios clientes. Por ejemplo, un cliente podría requerir solo tres vistas, y podría haber dos escenarios. En un ejemplo, un cliente puede requerir una experiencia de visualización sin problemas y podría preferir vistas con los valores del id_vista S0, S1 y S2, mientras que otro cliente puede requerir ajustabilidad a escala de las vistas y preferir vistas con los valores de id_vista S0, S2 y S4. Ambos sub-flujos de bits se pueden decodificar como flujos de bits independientes de MVC y pueden disponer de soporte de forma simultánea.

De acuerdo con las técnicas de esta divulgación, los datos de vídeo de una vista no de base se pueden predecir a partir de una vista de base. Además, un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede estar configurado para realizar la compensación de la iluminación al codificar muestras de luma de una vista no de base. Por ejemplo, como se muestra en la FIG. 4, los datos de la vista S2 pueden predecirse entre vistas a partir de los datos de la vista S0. Así, por ejemplo, un codificador de vídeo puede utilizar la compensación de iluminación a bloques de código de la imagen P en el momento T0 de la vista S2 con relación a los datos de la imagen I de vista S0 en el momento T0. A modo de ejemplo, el codificador de vídeo puede obtener las ponderaciones de predicción para una partición de bloque de vídeo de un bloque de vídeo (por ejemplo, un macro-bloque) de la imagen P en el momento T0 de la vista S2 una vez.

La FIG. 5 es un diagrama conceptual que ilustra la relación entre un bloque de una trama de referencia predicha entre vistas y un bloque actual de una trama decodificada. En el ejemplo de la FIG. 5, un bloque actual 100 de la trama decodificada 102 se relaciona con un bloque predictivo 104, predPart_i, de una trama de referencia entre vistas 106, mediante un vector de movimiento 108. El vector de movimiento 108 indica el desplazamiento relativo del

bloque actual 100 con relación al bloque predictivo 104.

A la izquierda, y encima, del bloque actual 100 y del bloque predictivo 104 hay bloques de muestras de luma. El bloque de muestras de luma contiguas izquierdas con respecto al bloque actual es $LTDec_i$ 110, y el bloque de muestras de luma contiguas superiores con respecto al bloque actual 100 es $UTDec_i$ 112. El bloque contiguo izquierdo en relación con el bloque predictivo 104 es $LTRef_i$ 114, y el bloque contiguo superior en relación con el bloque predictivo 104 es $UTRef_i$ 116.

La disposición de $UTRef_i$ 116, $LTRef_i$ 114, $UTDec_i$ 112 y $LTDec_i$ 110, ilustrada en la FIG. 5, es simplemente un ejemplo. De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar los tamaños y/o ubicaciones de $UTRef_i$, $LTRef_i$, $UTDec_i$, y $LTDec_i$, distintos a los ilustrados en la FIG. 5. Por ejemplo, $UTRef_i$, $LTRef_i$, $UTDec_i$ y $LTDec_i$ pueden tener dimensiones menores que la anchura o la altura del bloque actual 100 y el bloque predictivo 104. Aunque solo un vector de movimiento, el vector de movimiento 108, se ilustra en la FIG. 5, puede haber diferentes vectores de movimiento que indican las posiciones de una o más entre $LTRef_i$ 114 y $UTRef_i$ 116, como ejemplos.

Las FIGs. 6A a 6D son diagramas conceptuales que ilustran ejemplos de técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación. Como se ha descrito brevemente en lo que antecede, con respecto a la FIG. 1, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para utilizar bloques de regiones de muestras de luma contiguas superiores e izquierdas, de diferentes tamaños y ubicaciones. De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 de las FIGs. 1 a 3 pueden estar configurado para determinar bloques de regiones de muestras de luma con un tamaño de 8×4 muestras, para la región de muestras contiguas superiores, y de 4×8 muestras para la región contigua izquierda. La FIG. 6A ilustra una configuración en la que el codificador de vídeo 20 o el decodificador de vídeo 30 utiliza una muestra contigua superior de tamaño 8×4 , de acuerdo con las técnicas de esta divulgación.

La FIG. 6B ilustra otra posible configuración de bloques de regiones de muestras de luma de tamaño 4×8 o 8×4 , que el codificador de vídeo 20 o el decodificador de vídeo 30 puede ser configurado para seleccionar al realizar la ALC. En el ejemplo de la FIG. 6B, el codificador de vídeo 20 o el decodificador de vídeo 30 selecciona la región media de tamaño 8×4 para la muestra contigua superior, y puede seleccionar un bloque de regiones de muestras de luma centrales contiguas izquierdas de tamaño 4×8 (no se muestra en la FIG. 6B).

En algunos ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar regiones superiores cualesquiera de tamaño 8×4 dentro de la región superior original de tamaño 16×4 , con respecto al bloque actual cuando N es igual a 16. El codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar regiones cualesquiera de tamaño 4×8 dentro de la región original de tamaño 4×16 cuando M es igual a 16. En los casos en que los bloques de muestras de luma para la ponderación de la ALC son de muestras de tamaño 8×4 o 4×8 , el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar la ubicación de los uno o más bloques de referencia de tamaño 8×4 o 8×4 (es decir, $LTRef_i$ y $UTRef_i$) de varias formas, por ejemplo, similar a lo que se hace en el borrador de trabajo actual de la norma 3D-AVC.

La FIG. 6C ilustra otra posible configuración más de bloques de regiones de muestras de luma para el codificador de vídeo 20 o el decodificador de vídeo 30. En la FIG. 6C, si N, la anchura de la partición del macro-bloque actual, o M, la altura de la partición del macro-bloque actual, no es igual a 4, el codificador de vídeo 20 o el decodificador de vídeo 30 determina una región de muestras de luma contiguas superiores más a la izquierda, de tamaño 4×4 , para su uso al determinar factores de ponderación para la ALC. El codificador de vídeo 20 o el decodificador de vídeo 30 pueden estar configurados de manera similar para determinar una región de bloques de muestras contiguas izquierdas superiores centrales (no ilustrado en la FIG. 6C).

La FIG. 6D ilustra otra posible configuración de regiones de muestras de luma para el codificador de vídeo 20 o el decodificador de vídeo 30 al realizar la ALC. En el ejemplo de la FIG. 6D, si N o M no es igual a cuatro, el codificador de vídeo 20 o el decodificador de vídeo 30 está configurado para determinar un bloque de regiones de muestras de luma contiguas superiores centrales de tamaño 4×4 para su uso al realizar la ALC. De manera similar, en este ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 está configurado para determinar un bloque de muestras de luma contiguas izquierdas centrales de tamaño 4×4 , para su uso al realizar la ALC.

En otra alternativa, sin tener en cuenta el valor de N o M, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar la región de muestras de luma superior y la región de muestras de luma izquierda, como una región que tiene un tamaño de 4×4 muestras de luma. Como alternativa, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar cualquier región de tamaño 4×4 dentro de la región de tamaño $N \times 4$ y $4 \times M$.

Las FIGs. 7A a 7B son diagramas conceptuales que ilustran ejemplos de técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación. Como se ha descrito brevemente anteriormente, con respecto a la FIG. 1, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para realizar la identificación de muestras de referencia, dependiente de la partición de macro-bloque, pero independiente de la partición de sub-bloque. En estas técnicas, el codificador de vídeo 20 o el decodificador de vídeo 30 puede obtener

un conjunto de ponderaciones de la ALC para cada partición de macro-bloque. El codificador de vídeo 20 o el decodificador de vídeo 30 también pueden configurarse para filtrar conjuntamente píxeles dentro de una partición de macro-bloque utilizando el conjunto de ponderaciones, independientemente de la partición de sub-bloque. Por ejemplo, para una partición de macro-bloque de tamaño 8x8, si la partición de tamaño 8x8 se divide además en cuatro bloques de tamaño 4x4, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar el mismo conjunto de coeficientes de filtro para los cuatro bloques, y puede filtrar los bloques de forma conjunta.

En otro ejemplo ilustrado en la FIG. 7A, el codificador de vídeo 20 o el decodificador de vídeo 30 selecciona las regiones de muestras de luma contiguas superiores e izquierdas para el bloque superior de tamaño 4x4 y el bloque izquierdo de tamaño 4x4 del bloque superior izquierdo de tamaño 4x4 de una partición de macro-bloque. En este ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para utilizar el vector de movimiento de la región de muestras de luma contiguas superiores y/o izquierdas de tamaño 4x4, para determinar la ubicación del bloque predictivo de las muestras de luma a partir de la vista de referencia (por ejemplo, LTRef_i y UTRef_i).

En otra alternativa, si una partición de macro-bloque tiene un tamaño de MxN, y M o N es más grande que 8x8, o M o N tiene un tamaño igual a 8x8, pero tiene una partición de sub-bloque de tamaño 8x8, el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar la región de muestras de referencia de luma izquierda, de tal manera que la región tenga un tamaño de 4xN, y una región de muestras de referencia superior, de tal manera que la región de muestras de referencia superior tenga un tamaño de MX4. El codificador de vídeo 20 o el decodificador de vídeo 30 también puede seleccionar la región de muestras de referencia izquierda de tal manera que la región de muestras de referencia de luma izquierda tenga un tamaño de 4x8 y la región de muestras de referencia derecha de tal manera que la región de muestras de referencia de luma derecha tenga un tamaño de 8x4. El codificador de vídeo 20 o el decodificador de vídeo 30 de tal manera que las regiones de muestras de referencia de luma izquierda y superior tengan un tamaño de 4xN/2 y M/2x4, respectivamente.

En varios ejemplos, tales como los ilustrados en la FIG. 7B, el codificador de vídeo 20 o el decodificador de vídeo 30 puede realizar la identificación de muestras de referencia, dependiente de la partición de macro-bloque, y dependiente de la partición de sub-bloque. En el ejemplo de la FIG. 7B, el codificador de vídeo 20 o el decodificador de vídeo 30 obtienen un conjunto de ponderaciones de predicción de ALC para cada partición de MB. Sin embargo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar diferentes técnicas para identificar las muestras de referencia para cada partición de sub-bloque. Si la partición de macro-bloque es una partición de tamaño 8x8, y la partición de sub-bloque tiene un tamaño de: dos bloques de tamaño 8x4 (mxn) o dos bloques de tamaño 4x8 (mxn), el codificador de vídeo 20 o el decodificador de vídeo 30 puede seleccionar la región de muestras de referencia izquierda de modo que la región de muestras de referencia contigua izquierda tenga un tamaño de 4xn y las muestras de luma de referencia superior un tamaño de mx4.

En una alternativa, si la partición de macro-bloque es de tamaño 8x8 y la partición de sub-bloque es de dos bloques de tamaño 8x4 (mxn) o dos bloques de tamaño 4x8 (mxn), el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar el vector de movimiento indicativo de la ubicación del bloque predictivo de las muestras de luma para el bloque actual, sobre la base de uno de los vectores de movimiento indicativos de las ubicaciones de los bloques contiguos superiores o izquierdos de las muestras de luma de referencia. En otra alternativa, el codificador de vídeo 20 o el decodificador de vídeo 30 puede utilizar los vectores de movimiento que indican la ubicación de los bloques de muestras de referencia contiguas superiores y/o izquierdas, al determinar el vector de movimiento indicativo del bloque predictivo de las muestras de luma para el bloque actual.

Las técnicas descritas en las modificaciones al borrador de trabajo de la norma 3D-AVC incluyen técnicas que hacen que un codificador configurado de acuerdo con las técnicas de esta divulgación lleve a cabo la compensación de movimiento para determinar las ponderaciones para la ALC solo una vez por partición de macro-bloque. El codificador de vídeo utiliza luego el mismo vector de movimiento para determinar las muestras de luma de referencia para cada partición de macro-bloque y sub-particiones cualesquiera dentro de la partición de macro-bloque.

La FIG. 8 es un diagrama de flujo que ilustra una operación a modo de ejemplo de un dispositivo de codificación de vídeo configurado para realizar técnicas de compensación de luminancia adaptativa de acuerdo con las técnicas de esta divulgación. Para los fines de este ejemplo, se debería entender que un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede llevar a cabo el procedimiento de la FIG. 8.

En el procedimiento de la FIG. 8, el codificador de vídeo 20 o el decodificador de vídeo 30 puede obtener las ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición de bloque de vídeo, una vez por partición de bloque de vídeo, de tal manera que la partición del bloque de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente del tamaño de transformación de las muestras de luma para la partición del bloque de vídeo (160).

En el procedimiento de la FIG. 8, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para calcular un bloque predicho para la partición de bloques de vídeo, utilizando las ponderaciones de

predicción y usando la compensación de la iluminación (162), y para codificar la partición de bloques de vídeo usando el bloque predicho (164).

5 Para obtener el conjunto común de ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede además estar configurado para obtener el conjunto común de ponderaciones de predicción usando muestras de luma de a lo sumo $N \times 4$ muestras de un bloque contiguo superior, y para usar muestras de luma de a lo sumo $4 \times M$ muestras de un bloque contiguo izquierdo. En este ejemplo, N es un ancho y M es una altura, un N y M son menores o iguales a 16.

10 En varios ejemplos, el bloque contiguo superior tiene una altura de 1 fila de muestras de luma, y el bloque contiguo izquierdo tiene una anchura de 1 columna de muestras de luma. En algunos ejemplos, el bloque contiguo superior puede incluir solo un primer subconjunto de muestras de luma de la fila 1 de muestras de luma, y el bloque contiguo izquierdo puede incluir solo un segundo subconjunto de muestras de luma de la columna 1 de muestras de luma.

15 En algunos ejemplos, el bloque contiguo superior incluye solo las muestras de luma de los extremos izquierdo y derecho de las muestras de luma, y el bloque contiguo izquierdo incluye solo las muestras de luma de los extremos superior e inferior de la columna 1 de muestras de luma. En algunos otros ejemplos, uno entre N y M es uno entre 4, 8 o 16, y el otro entre N y M es igual a 16.

20 En otro ejemplo más, la partición de bloques de vídeo puede tener un tamaño igual a $N \times M$, donde N es una anchura, y M es una altura. En este ejemplo, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden obtener las ponderaciones de predicción utilizando al menos uno entre: una serie de columnas de píxeles en un bloque contiguo superior basado en N , y una serie de filas de píxeles en un bloque contiguo izquierdo basado en M .

25 En otros ejemplos más, las $N \times 4$ muestras del bloque contiguo izquierdo pueden comprender muestras de luma contiguas izquierdas cualesquiera de tamaño 4×8 , en relación con la partición del bloque de vídeo. En otro ejemplo, las $N \times 4$ muestras del bloque contiguo izquierdo pueden comprender muestras de luma contiguas izquierdas cualesquiera de tamaño 4×4 , en relación con la partición del bloque de vídeo. En otro ejemplo, las $4 \times M$ muestras del bloque contiguo superior pueden comprender muestras de luma contiguas izquierdas cualesquiera de tamaño 8×4 , en relación con la partición del bloque de vídeo. En otro ejemplo, las $4 \times M$ muestras del bloque contiguo superior pueden comprender muestras de luma contiguas izquierdas cualesquiera de tamaño 4×4 , en relación con el bloque de la partición de vídeo. En otros ejemplos más aún, si al menos uno entre N y M no es igual a 4, el bloque contiguo superior es el bloque contiguo superior más a la izquierda, y el bloque contiguo izquierdo es el bloque contiguo izquierdo más elevado.

35 En otros ejemplos más aún, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para obtener las ponderaciones de predicción usando muestras finales y una muestra de centro de la fila de muestras o la columna de muestras. En otro ejemplo, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para obtener las ponderaciones de predicción utilizando a lo sumo cuatro muestras de la fila de muestras o la columna de muestras.

40 En varios ejemplos, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para obtener las ponderaciones de predicción utilizando uno o más muestras de muestreo reducido. En otro ejemplo, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para obtener las ponderaciones de predicción utilizando uno o más muestras de muestreo reducido y filtradas.

45 En otros ejemplos, si la partición tiene un tamaño de 8×8 , donde el conjunto común de ponderaciones de predicción comprende un único conjunto de ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede además estar configurado para obtener el único conjunto de ponderaciones de predicción para la compensación de la iluminación de la partición del bloque de vídeo en base a ponderaciones de predicción para un bloque de muestras de luma de la partición del bloque de vídeo.

50 En otro ejemplo, la partición de tamaño 8×8 de muestras de luma puede comprender además al menos una sub-partición de muestras de luma de tamaño 4×4 , y el bloque utilizado para obtener las ponderaciones de predicción puede ser una de las al menos una sub-partición de muestras de luma de tamaño 4×4 . En diversos ejemplos, la partición de tamaño 8×8 de muestras de luma comprende además al menos una sub-partición de las muestras de luma de tamaño 4×4 . En estos ejemplos, el codificador de vídeo 20 o el decodificador 30 de vídeo puede además ser configurado para seleccionar, como el bloque utilizado para obtener las ponderaciones de predicción, una de las al menos una sub-partición de muestras de luma de tamaño 4×4 que tenga un máximo vector de movimiento entre los vectores de movimiento para las sub-particiones.

55 En otro ejemplo, el conjunto de ponderaciones de predicción se basa en un conjunto de muestras de luma contiguas izquierdas previamente decodificadas, y un conjunto de muestras de luma contiguas superiores con respecto a la

partición de bloques de vídeo en la misma imagen que la partición del bloque de vídeo, y en el que el conjunto de ponderaciones de predicción también se basa en un conjunto de muestras de luma contiguas izquierdas y un conjunto de muestras de luma contiguas superiores con relación a un bloque de referencia predictivo entre vistas de las muestras de luma para la partición de bloques de vídeo.

5 En otro ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30 puede además ser configurado para: determinar un vector de movimiento entre vistas para la partición de bloques de vídeo, redondear el vector de movimiento entre vistas hasta un píxel entero y determinar el bloque de referencia predictivo entre vistas de las muestras de luma en base al vector de movimiento redondeado a mitades de enteros.

10 En otros ejemplos más, el codificador de vídeo 20 o el decodificador de vídeo 30 puede además estar configurado para: determinar un vector de movimiento entre vistas para la partición de bloques de vídeo. El codificador de vídeo 20 o el decodificador de vídeo 30 puede redondear el vector de movimiento entre vistas hasta mitades de enteros de píxeles, y determinar el bloque de referencia predictivo entre vistas de las muestras de luma en base al vector de movimiento redondeado hasta mitades de enteros.

15 En otros ejemplos más, el bloque de referencia predictivo entre vistas puede comprender un primer bloque, y el conjunto de muestras de luma contiguas izquierdas y el conjunto de muestras de luma contiguas superiores, con respecto a un segundo bloque, diferente, de referencia entre vistas, determinado en base a un vector de movimiento redondeado que indica la ubicación del primer bloque.

20 En otros ejemplos más, la partición de bloques de vídeo comprende una partición de bloque de vídeo entre una o más particiones de un bloque de vídeo completo y, para obtener el conjunto común de ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 está configurado para obtener el conjunto común de ponderaciones de predicción una vez para todas las particiones de bloque de vídeo en base a aquella entre las particiones de bloques de vídeo que contiene un píxel superior izquierdo del bloque de vídeo completo.

25 En otro ejemplo, la partición del bloque de vídeo puede comprender una pluralidad de sub-bloques, y cada uno de los sub-bloques puede tener un vector de movimiento respectivo. Para obtener el conjunto común de ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 está configurado para obtener el conjunto común de ponderaciones de predicción de una vez para cada uno de los sub-bloques, en base al sub-bloque superior izquierdo de la partición del bloque de vídeo.

30 En otro ejemplo más, la partición de bloques de vídeo puede tener un tamaño igual a $N \times M$, y al menos uno entre N y M puede ser igual a 16. En este ejemplo, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 está configurado para obtener las ponderaciones de predicción utilizando hasta N columnas de píxeles en un bloque contiguo superior o hasta M filas de píxeles en un bloque contiguo izquierdo.

35 En algunos ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado además para llevar a cabo una operación de filtrado en las hasta N columnas de píxeles o las hasta M filas de píxeles para formar un conjunto de muestras intermedias. En estos ejemplos, para obtener las ponderaciones de predicción, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para obtener las ponderaciones de predicción en base, al menos en parte, a las muestras intermedias. Y en algunos ejemplos, para llevar a cabo la operación de filtrado, el codificador de vídeo 20 o el decodificador de vídeo 30 puede estar configurado para realizar la operación de filtrado en muestras dentro de las hasta N columnas o las hasta M filas que pueden formar una región de tamaño 8×4 , 4×8 o 4×4 .

40 De esta manera, el procedimiento de la FIG. 8 representa un ejemplo de un procedimiento que incluye obtener ponderaciones de predicción para la compensación de la iluminación de las muestras de luma de una partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción para realizar la compensación de la iluminación de las muestras de luma, independientemente de un tamaño de transformación para la partición de bloques de vídeo, calcular un bloque predicho para la partición de bloques de vídeo utilizando las ponderaciones de predicción y utilizando la compensación de la iluminación, y codificar (por ejemplo, codificar o decodificar) la partición de bloques de vídeo usando el bloque predicho.

45 Si bien se describen ciertas técnicas de esta divulgación con respecto a la norma H.264, se debería entender que las técnicas no se limitan necesariamente a una norma de codificación particular. Es decir, las técnicas se refieren más en general al logro de eficacias de codificación en la 3DVC, por ejemplo, mediante un procesamiento más eficaz (por ejemplo, la codificación) de datos de vídeo, como se ha descrito anteriormente.

50 Ha de reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de los procedimientos descritos en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fundirse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica del procedimiento). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hebras, el procesamiento de interrupciones o múltiples procesadores, en

lugar de secuencialmente. Además, aunque ciertos aspectos de esta divulgación se describen como realizados por un único módulo o unidad, con fines de claridad, se debería entender que las técnicas de esta divulgación pueden ser realizadas por una combinación de unidades o módulos asociados con un codificador de vídeo.

5 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, como una o más instrucciones o código, pueden almacenarse en, o transmitirse por, 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
10 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 tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera, a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

20 A modo de ejemplo, y no de manera limitativa, tales 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. Además, cualquier conexión puede denominarse adecuadamente un
25 medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, 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 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.

30 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 medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

40 Las instrucciones pueden ser ejecutadas por 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), formaciones lógicas programables en el terreno (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior 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
45 proporcionarse dentro de hardware especializado y/o módulos de software configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

50 Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un equipo de mano inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden combinarse diversas unidades en una unidad de hardware de códec, o ser proporcionadas por una colección de
55 unidades de hardware inter-operativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con el software y/o firmware adecuado.

Se han descrito diversos aspectos de la divulgación. Estos y otros aspectos están dentro del alcance de las siguientes reivindicaciones.

60

REIVINDICACIONES

1. Un procedimiento de decodificación de datos de vídeo, procedimiento que comprende:
 - 5 obtener (160) las ponderaciones de predicción, en base a un tamaño de una partición de bloques de vídeo, y no en base a un tamaño de transformación para la partición de bloques de vídeo, para la compensación de la iluminación de las muestras de luma de la partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción entre las muestras de luma para realizar la
 - 10 compensación de la iluminación de las muestras de luma;
 - calcular (162) un bloque predicho para la partición de bloques de vídeo utilizando las ponderaciones de predicción y utilizando la compensación de la iluminación; y
 - 15 decodificar (164) la partición de bloques de vídeo usando el bloque predicho.

2. Un procedimiento de codificación de datos de vídeo, procedimiento que comprende:
 - 20 obtener (160) las ponderaciones de predicción, en base a un tamaño de una partición de bloques de vídeo, y no en base a un tamaño de transformación para la partición de bloques de vídeo, para la compensación de la iluminación de las muestras de luma de la partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción entre las muestras de luma para realizar la
 - 25 compensación de la iluminación de las muestras de luma;
 - calcular (162) un bloque predicho para la partición de bloques de vídeo utilizando las ponderaciones de predicción con compensación de la iluminación; y
 - 30 codificar (164) la partición de bloques de vídeo usando el bloque predicho.

3. El procedimiento de la reivindicación 1 o 2, en el que la obtención del conjunto común de ponderaciones de predicción comprende obtener el conjunto común de ponderaciones de predicción usando muestras de luma de a lo sumo $N \times 4$ muestras de un bloque contiguo superior, y el uso de muestras de luma de a lo sumo $4 \times M$ muestras de un bloque contiguo izquierdo,
 - 35 en el que N es un ancho y M es una altura, y en el que N y M son menores o iguales a 16.

4. El procedimiento de la reivindicación 3,
 - 40 en el que el bloque contiguo superior tiene una altura de 1 fila de muestras de luma, en el que el bloque contiguo izquierdo tiene una anchura de 1 columna de muestras de luma;
 - en el que el bloque contiguo superior incluye solo un primer subconjunto de muestras de luma de la fila 1 de muestras de luma, y en el que el bloque contiguo izquierdo solo incluye un segundo subconjunto de las muestras de luma de la
 - 45 columna 1 de las muestras de luma.

5. El procedimiento de la reivindicación 3, en el que:
 - 50 las $N \times 4$ muestras del bloque contiguo izquierdo comprenden muestras de luma contiguas izquierdas cualesquiera de tamaño 4×8 , en relación con la partición de bloques de vídeo; o
 - las $N \times 4$ muestras del bloque contiguo izquierdo comprenden muestras de luma contiguas izquierdas cualesquiera de tamaño 4×4 , en relación con la partición de bloques de vídeo; o
 - 55 las $4 \times M$ muestras del bloque contiguo superior comprenden muestras de luma contiguas izquierdas cualesquiera de tamaño 8×4 , en relación con la partición de bloques de vídeo; o
 - las $4 \times M$ muestras del bloque contiguo superior comprenden muestras de luma contiguas izquierdas cualesquiera de tamaño 4×4 , en relación con la partición de bloques de vídeo.

6. El procedimiento de la reivindicación 2 o la reivindicación 3,
 - 60 en el que al menos uno entre N y M no es igual a 4, en el que el bloque contiguo superior es el bloque contiguo superior más a la izquierda, y en el que el bloque contiguo izquierdo es el bloque contiguo izquierdo más alto.

7. El procedimiento de la reivindicación 3, en el que la obtención de las ponderaciones de predicción

comprende:

- 5 i) obtener las ponderaciones de predicción usando muestras de los extremos izquierdo y derecho y un centro del bloque contiguo superior, y muestras de los extremos superior e inferior, y un centro del bloque contiguo izquierdo; u
- 10 ii) obtener las ponderaciones de predicción comprende obtener las ponderaciones de predicción utilizando a lo sumo cuatro muestras de la fila de muestras o la columna de muestras; u
- 15 iii) obtener las ponderaciones de predicción comprende obtener las ponderaciones de predicción utilizando una o más muestras de muestreo reducido; u
- 15 iv) obtener las ponderaciones de predicción comprende obtener las ponderaciones de predicción utilizando una o más muestras de muestreo reducido y filtradas.
8. El procedimiento de la reivindicación 1 o 2, en el que la partición tiene un tamaño de 8x8, en el que el conjunto común de ponderaciones de predicción comprende un único conjunto de ponderaciones de predicción, comprendiendo el procedimiento además obtener el único conjunto de ponderaciones de predicción para la compensación de la iluminación de la partición del bloque de vídeo en base a ponderaciones de predicción para un bloque de muestras de luma de la partición de bloques de vídeo.
- 20 9. El procedimiento de la reivindicación 1 o 2, en el que el conjunto de ponderaciones de predicción se basa en un conjunto de muestras de luma contiguas izquierdas previamente decodificadas, y un conjunto de muestras de luma contiguas superiores, en relación con la partición de bloques de vídeo en la misma imagen que la partición de bloques de vídeo, y en el que el conjunto de ponderaciones de predicción también se basa en un conjunto de muestras de luma contiguas izquierdas y un conjunto de muestras de luma contiguas superiores, con relación a un bloque de referencia predictivo entre vistas de las muestras de luma para la partición de bloques de vídeo.
- 25 10. El procedimiento de la reivindicación 9, en el que el bloque de referencia predictivo entre vistas comprende un primer bloque, en el que el conjunto de muestras de luma contiguas izquierdas y el conjunto de muestras de luma contiguas superiores son contiguos respecto a un segundo bloque de referencia, distinto, entre vistas, determinado en base a un vector de movimiento redondeado que indica la ubicación del primer bloque.
- 30 11. El procedimiento de la reivindicación 1 o 2, en el que:
- 35 i) la partición de bloques de vídeo comprende una partición de bloques de vídeo de una o más particiones de bloques de vídeo de un bloque de vídeo completo, y en el que la obtención del conjunto común de ponderaciones de predicción comprende obtener el conjunto común de ponderaciones de predicción una vez para todas las particiones de bloques de vídeo, en base a aquella de las particiones de bloques de vídeo que contiene un píxel superior izquierdo del bloque de vídeo completo; o
- 40 ii) la partición de bloques de vídeo comprende una pluralidad de sub-bloques, teniendo cada uno de los sub-bloques un vector de movimiento respectivo, y en el que la obtención del conjunto común de ponderaciones de predicción comprende obtener el conjunto común de ponderaciones de predicción, una vez para cada uno de los sub-bloques, en base al sub-bloque superior izquierdo de la partición de bloques de vídeo.
- 45 12. El procedimiento de la reivindicación 1 o 2, en el que la partición de bloques de vídeo tiene un tamaño igual a NxM, en la que N es una anchura, y M es una altura, y donde obtener las ponderaciones de predicción comprende obtener las ponderaciones de predicción utilizando al menos uno entre: una serie de columnas de píxeles en un bloque contiguo superior, sobre la base de N, y un número de filas de píxeles en un bloque contiguo izquierdo, basado en M.
- 50 13. Un medio de almacenamiento no transitorio legible por ordenador que comprende instrucciones que, cuando son ejecutadas, hacen que uno o más procesadores lleven a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 12.
- 55 14. Un dispositivo para decodificar datos de vídeo (30), dispositivo que comprende:
- 60 medios para obtener ponderaciones de predicción, en base a un tamaño de una partición de bloques de vídeo, y no en base a un tamaño de transformación para la partición de bloques de vídeo, para la compensación de la iluminación de las muestras de luma de la partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un
- 65

conjunto común de ponderaciones de predicción entre las muestras de luma para realizar la compensación de la iluminación de las muestras de luma;

5 medios para calcular un bloque predicho para la partición de bloques de vídeo, utilizando las ponderaciones de predicción usando la compensación de la iluminación; y

medios para decodificar la partición de bloques de vídeo usando el bloque predicho.

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

10 medios para obtener ponderaciones de predicción, en base a un tamaño de una partición de bloques de vídeo, y no en base a un tamaño de transformación para la partición de bloques de vídeo, para la compensación de la iluminación de las muestras de luma de la partición de bloques de vídeo, una vez para la partición de bloques de vídeo, de tal manera que la partición de bloques de vídeo tenga un conjunto común de ponderaciones de predicción entre las muestras de luma para realizar la compensación de la iluminación de las muestras de luma;

15 medios para calcular un bloque predicho para la partición de bloques de vídeo utilizando las ponderaciones de predicción y usando la compensación de la iluminación; y

20 medios para codificar la partición de bloques de vídeo usando el bloque predicho.

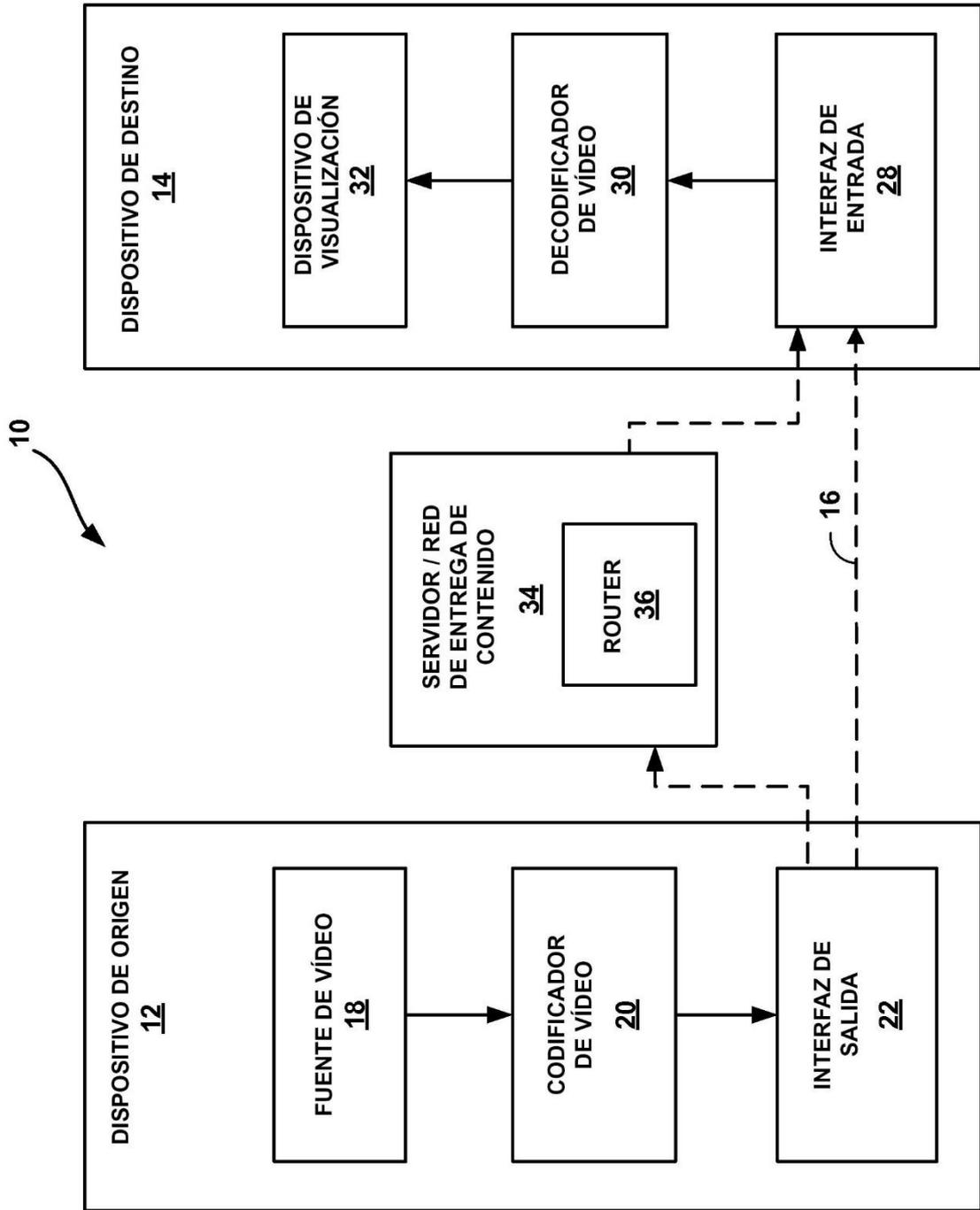


FIG. 1

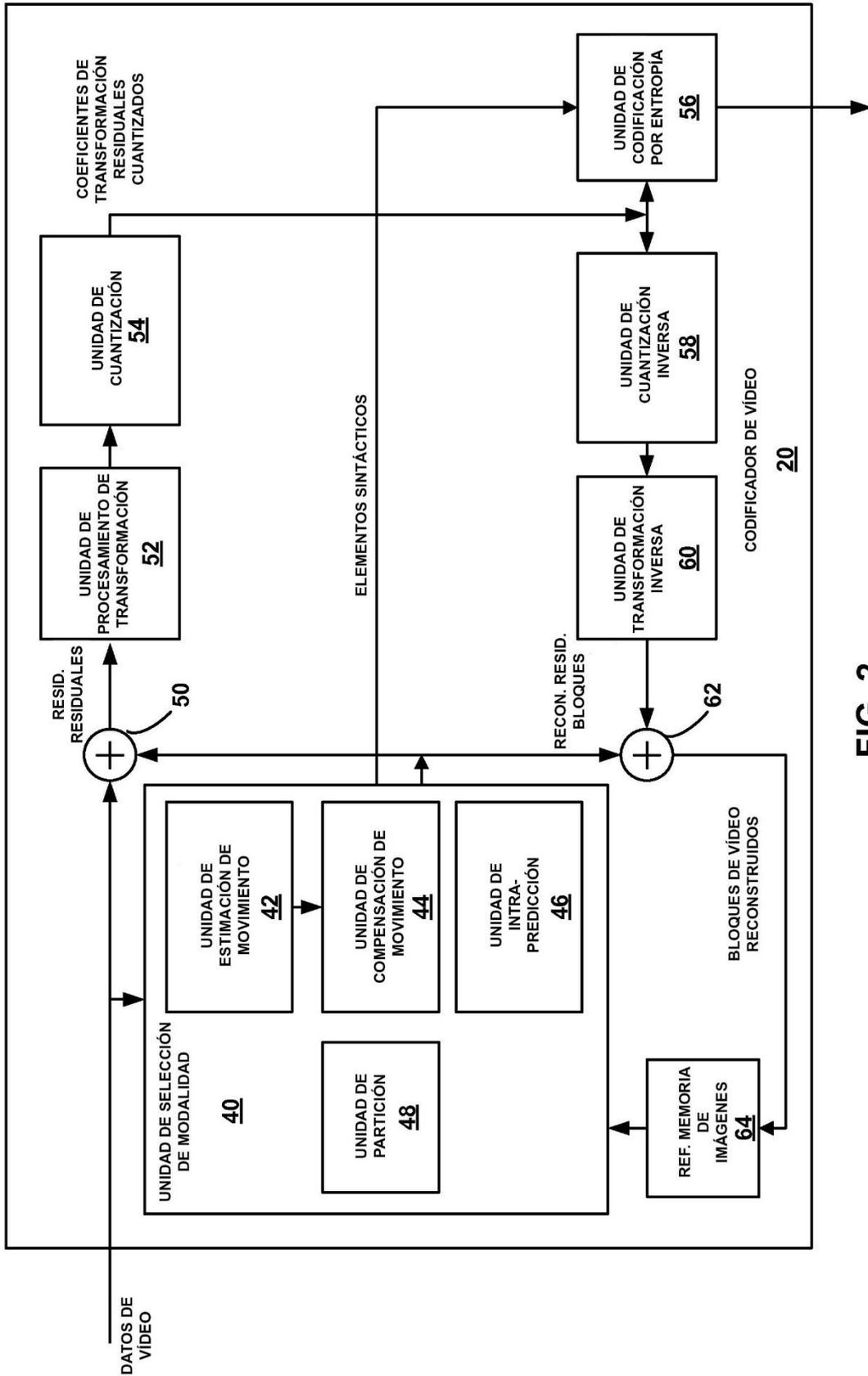


FIG. 2

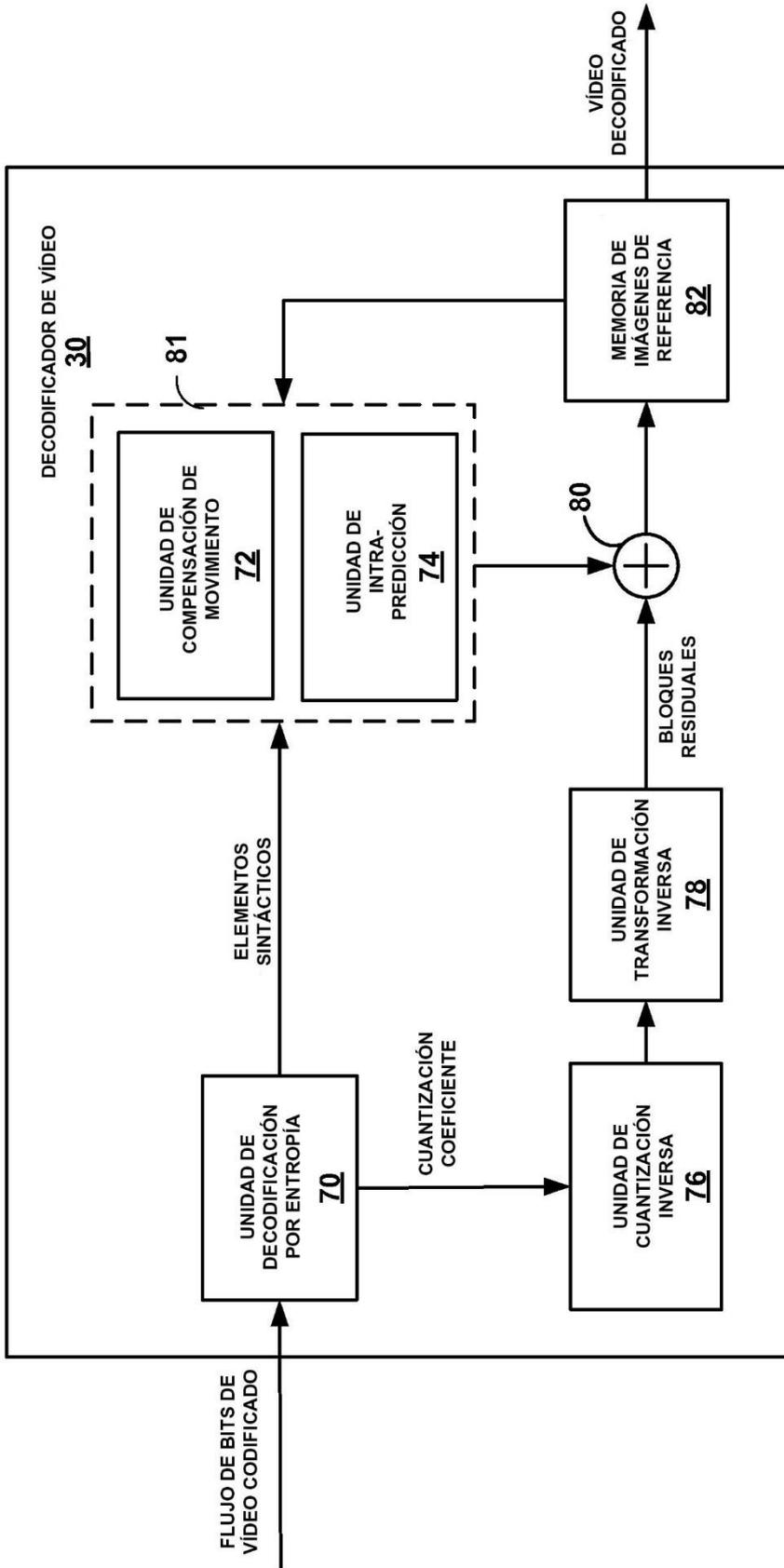


FIG. 3

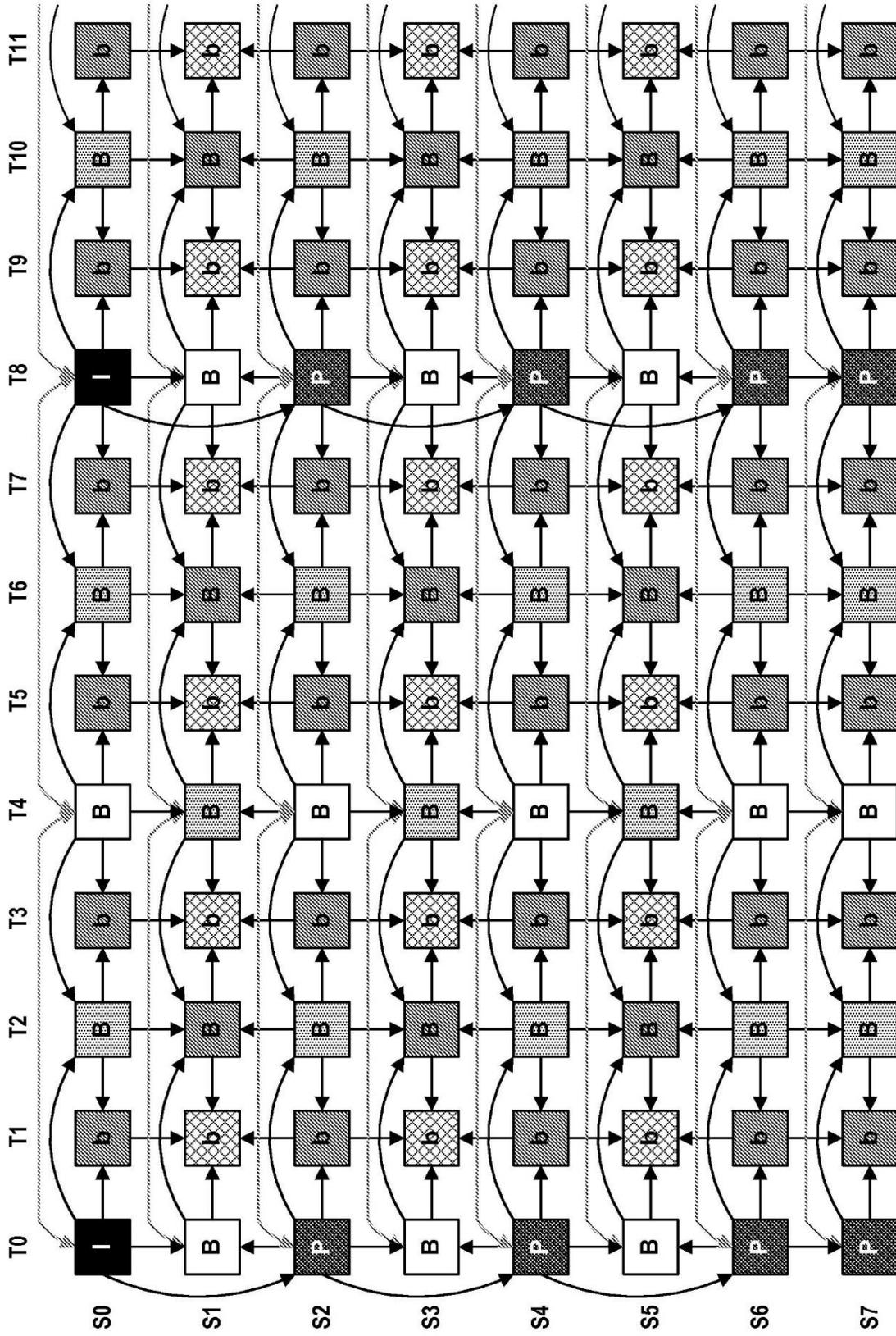


FIG. 4

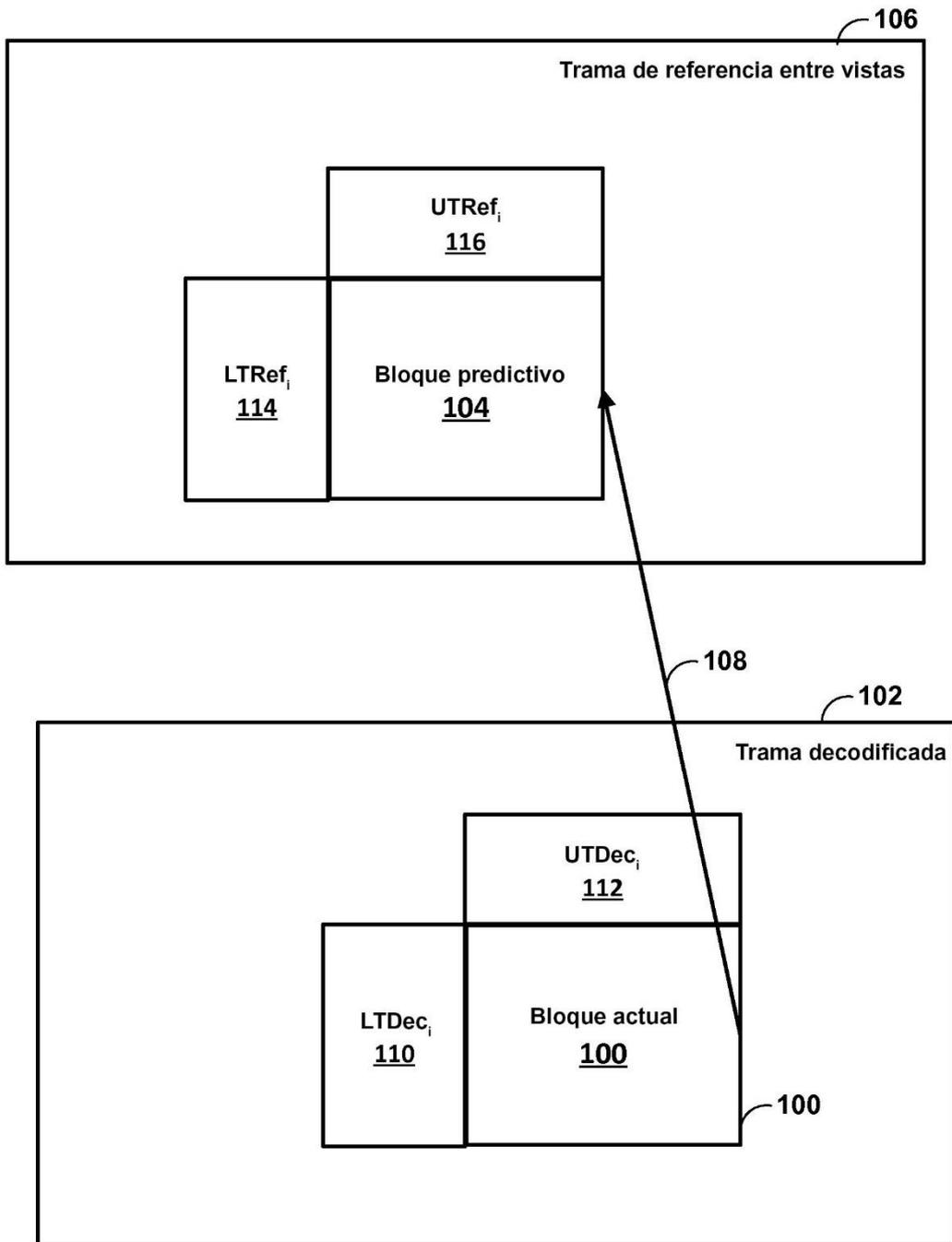


FIG. 5

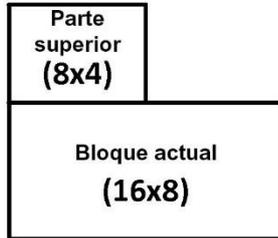


FIG. 6A

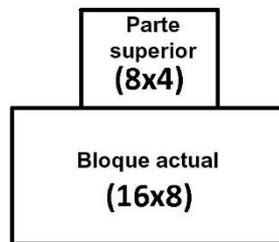


FIG. 6B

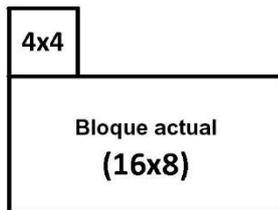


FIG. 6C

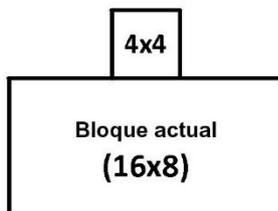


FIG. 6D

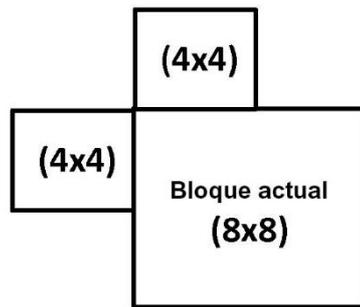


FIG. 7A

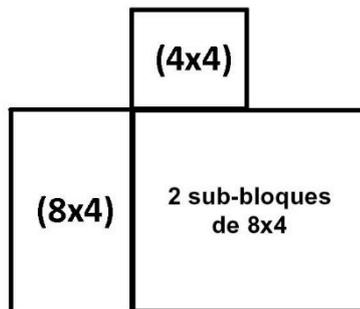


FIG. 7B

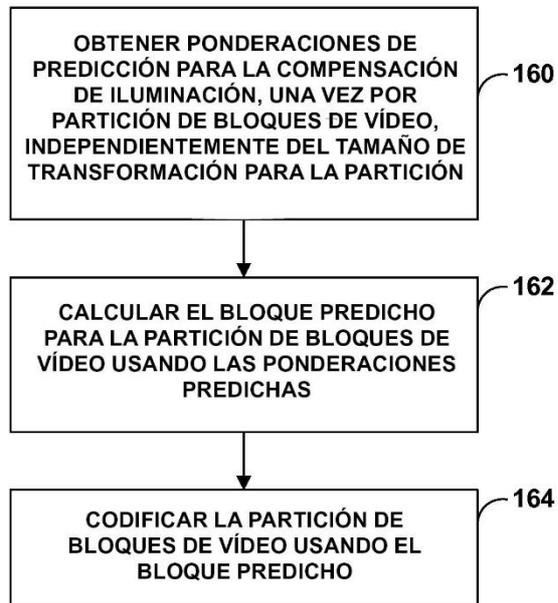


FIG. 8