

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 778 350**

51 Int. Cl.:

H04N 19/187	(2014.01)
H04N 19/33	(2014.01)
H04N 19/34	(2014.01)
H04N 19/42	(2014.01)
H04N 19/59	(2014.01)
H04N 19/597	(2014.01)
H04N 19/46	(2014.01)
H04N 19/136	(2014.01)
H04N 19/132	(2014.01)
H04N 19/117	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **12.03.2014 PCT/US2014/024423**
- 87 Fecha y número de publicación internacional: **25.09.2014 WO14150864**
- 96 Fecha de presentación y número de la solicitud europea: **12.03.2014 E 14716140 (0)**
- 97 Fecha y número de publicación de la concesión europea: **18.12.2019 EP 2974312**

54 Título: **Dispositivo y procedimiento para codificación escalable de información de vídeo**

30 Prioridad:

15.03.2013 US 201361790538 P
11.03.2014 US 201414205006

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
10.08.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
Attn: International IP Administration, 5775
Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

GUO, LIWEI;
CHEN, JIANLE;
LI, XIANG;
RAPAKA, KRISHNAKANTH;
PU, WEI y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 778 350 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

Dispositivo y procedimiento para codificación escalable de información de vídeo

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere al campo de la codificación y compresión de vídeo, en particular a la codificación de vídeo escalable (SVC) o la codificación de vídeo multivista (MVC, 3DV).

10 **ANTECEDENTES**

[0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, dispositivos de teleconferencia 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 de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

[0003] Las técnicas de compresión de vídeo realizan una predicción espacial (intraimagen) y/o una predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un sector de vídeo (por ejemplo, una trama de vídeo, una parte de una trama de vídeo, etc.) se puede dividir en bloques de vídeo, que también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un sector intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia en bloques vecinos de la misma imagen. Los bloques de vídeo de un sector intercodificado (P o B) de una imagen pueden usar una predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o una predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio del píxel hasta un dominio de la transformada, dando como resultado unos coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación entrópica para lograr aún más compresión.

45 **SUMARIO**

[0005] La codificación de vídeo escalable (SVC) se refiere a la codificación de vídeo en la que se usa una capa base (BL), a veces denominada capa de referencia (RL), y una o más capas de mejora escalables (EL). En la SVC, la capa base puede transportar datos de vídeo con un nivel de calidad básico. La una o más capas de mejora pueden transportar datos de vídeo adicionales para admitir, por ejemplo, niveles espaciales, temporales y/o de señal-ruido (SNR) más altos. Las capas de mejora se pueden definir en relación con una capa codificada previamente. Por ejemplo, una capa inferior puede servir como una BL, mientras que una capa superior puede servir como una EL. Las capas intermedias pueden servir como EL o RL, o como ambas. Por ejemplo, una capa del medio puede ser una EL para las capas de debajo de esta, tales como la capa base o cualquier capa de mejora intermedia, y al mismo tiempo servir como RL para una o más capas de mejora encima de esta. De forma similar, en la ampliación multivista o 3D de la norma HEVC, puede haber múltiples vistas, y la información de una vista se puede utilizar para codificar (por ejemplo, codificar o descodificar) la información de otra vista (por ejemplo, estimación de movimiento, predicción de vector de movimiento y/u otras redundancias).

[0006] En la SVC, se puede predecir un bloque actual de la capa de mejora usando la información de píxeles de la capa base. Por ejemplo, en un modo de codificación para la capa de mejora denominado modo intra BL, la textura (por ejemplo, valores de píxel o de muestra) de un bloque actual en la capa de mejora se puede predecir usando la textura de un bloque colocalizado en la capa base (el término "colocalizado" como se usa en la presente divulgación se puede referir a un bloque de otra capa que corresponde a la misma imagen que el bloque actual, por ejemplo, el bloque que se está prediciendo actualmente). Por tanto, en lugar de transmitir la textura del bloque

actual, el codificador de vídeo puede transmitir solo la diferencia (por ejemplo, el residuo) entre la textura del bloque actual y la textura del bloque de capa base colocalizado.

[0007] Sin embargo, para determinados sistemas de escalabilidad tales como la escalabilidad espacial y la escalabilidad de profundidad de bits, puede ser necesario modificar la información de píxeles de capa base antes de usarla para predecir la información de píxeles de capa de mejora. Por ejemplo, en la escalabilidad espacial, puede ser necesario sobremuestrear [upsample] la información de píxeles de capa base (por ejemplo, de acuerdo con la relación de resolución) antes de usarla para predecir la información de píxeles de capa de mejora, y en la escalabilidad de profundidad de bits, puede ser necesario someter la información de píxeles de capa base a una conversión de profundidad de bits (por ejemplo, con un desplazamiento de bits igual a la diferencia de profundidad de bits) antes de usarla para predecir la información de píxeles de la capa de mejora. Cuando la escalabilidad espacial y la escalabilidad de profundidad de bits están presentes, puede ser necesario sobremuestrear y convertir la información de píxeles de capa base hasta una profundidad de bits diferente antes de usarla para predecir la información de píxeles de capa de mejora. Sin embargo, realizar el sobremuestreo y la conversión de profundidad de bits en dos procesos separados puede dar como resultado una eficacia de codificación y/o una exactitud de predicción (por ejemplo, error de redondeo) reducidas. Por lo tanto, combinando el proceso de sobremuestreo y el proceso de conversión de profundidad de bits en un proceso de una sola fase, la eficacia de codificación se puede mejorar y la complejidad computacional se puede reducir.

[0008] Cada uno de los sistemas, procedimientos y dispositivos de esta divulgación tiene varios aspectos innovadores, ninguno de los cuales es el único responsable de los atributos deseables divulgados en el presente documento.

[0009] La invención se define mediante las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0010]

La **FIG. 1** es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación.

La **FIG. 2A** es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación.

La **FIG. 2B** es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación.

La **FIG. 3A** es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación.

La **FIG. 3B** es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación.

La **FIG. 4** es un diagrama conceptual que ilustra unas escalabilidades de SVC en diferentes dimensiones.

La **FIG. 5** es un diagrama conceptual que ilustra un ejemplo de estructura de un flujo de bits SVC.

La **FIG. 6** es un diagrama conceptual que ilustra unas unidades de acceso en un flujo de bits SVC.

La **FIG. 7** es un diagrama de conceptual que ilustra un ejemplo de predicción entre capas de acuerdo con un modo de realización de la presente divulgación.

La **FIG. 8** es un diagrama de flujo que ilustra un procedimiento de codificación de información de vídeo, de acuerdo con un modo de realización de la presente divulgación.

La **FIG. 9** es un diagrama de flujo que ilustra un procedimiento de codificación de información de vídeo, de acuerdo con otro modo de realización de la presente divulgación.

DESCRIPCIÓN DETALLADA

[0011] Determinados modos de realización descritos en el presente documento se refieren a la predicción entre capas para la codificación de vídeo escalable en el contexto de los códecs de vídeo avanzados, tal como la HEVC (codificación de vídeo de alta eficacia). Más específicamente, la presente divulgación se refiere a sistemas y procedimientos para mejorar el rendimiento de la predicción entre capas en la ampliación de codificación de vídeo escalable (SVC) de la HEVC.

[0012] En la descripción siguiente, se describen técnicas H.264/AVC relacionadas con determinados modos de realización y también se analizan la norma HEVC y unas técnicas relacionadas. Aunque en el presente documento se describen determinados modos de realización en el contexto de las normas HEVC y/o H.264, un experto en la técnica puede apreciar que los sistemas y procedimientos divulgados en el presente documento pueden ser aplicables a cualquier norma de codificación de vídeo adecuada. Por ejemplo, los modos de realización divulgados en el presente documento pueden ser aplicables a una o más de las siguientes normas: ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC).

[0013] La HEVC, en general, sigue el marco de las normas de codificación de vídeo previas en muchos aspectos. La unidad de predicción en HEVC es diferente de la de determinadas normas de codificación de vídeo previas (por ejemplo, un macrobloque). De hecho, el concepto de macrobloque no existe en HEVC como se entiende en determinadas normas de codificación de vídeo previas. Un macrobloque se reemplaza por una estructura jerárquica basada en un sistema de árbol cuaternario, lo cual puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, dentro del sistema HEVC, se definen tres tipos de bloques: unidad de codificación (CU), unidad de predicción (PU) y unidad de transformada (TU). La CU se puede referir a la unidad básica de división de una región. La CU se puede considerar análoga al concepto de macrobloque, pero no limita el tamaño máximo y puede permitir una división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. La PU se puede considerar la unidad básica de inter/intrapredicción, y puede contener múltiples divisiones de forma arbitraria en una única PU para codificar eficazmente patrones de imagen irregulares. La TU se puede considerar la unidad básica de transformada. Se puede definir independientemente de la PU; sin embargo, su tamaño puede estar limitado a la CU a la que pertenece la TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir optimizar cada uno de ellos de acuerdo con su función, lo cual puede dar como resultado una mejora de la eficacia de codificación.

[0014] Para propósitos de ilustración solo, determinados modos de realización divulgados en el presente documento se describen con ejemplos que incluyen solo dos capas (por ejemplo, una capa de nivel inferior tal como la capa base y una capa de nivel superior tal como la capa de mejora). Se debe entender que dichos ejemplos pueden ser aplicables a configuraciones que incluyen múltiples capas base y/o de mejora. Además, para facilitar la explicación, la siguiente divulgación incluye los términos "tramas" o "bloques" con referencia a determinados modos de realización. Sin embargo, estos términos no pretenden ser limitativos. Por ejemplo, las técnicas descritas a continuación se pueden usar con cualquier unidad de vídeo adecuada, tal como un bloque (por ejemplo, una CU, una PU, una TU, un macrobloque, etc.), sector, trama, etc.

Normas de codificación de vídeo

[0015] Una imagen digital, tal como una imagen de vídeo, una imagen de TV, una imagen fija o una imagen generada por una grabadora de vídeo o un ordenador, puede consistir en píxeles o muestras dispuestas en líneas horizontales y verticales. El número de píxeles de una sola imagen suele ser de decenas de miles. Cada píxel contiene típicamente información de luminancia y crominancia. Sin compresión, la cantidad de información que se va a transmitir desde un codificador de imágenes hasta un descodificador de imágenes es tan enorme que haría imposible la transmisión de imágenes en tiempo real. Para reducir la cantidad de información que se va a transmitir, se han elaborado un número de procedimientos de compresión diferentes, tales como las normas JPEG, MPEG y H.263.

[0016] Entre las normas de codificación de vídeo se incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC).

[0017] Además, existe una nueva norma de codificación de vídeo, a saber, la codificación de vídeo de alta eficacia (HEVC), que el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC están elaborando. La cita completa del HEVC Draft 10 es el documento JCTVC-L1003, de Bross *et al.*, "High Efficiency Video Coding (HEVC) Text Specification Draft 10", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12.^a reunión: Ginebra, Suiza, 14 de enero de 2013 a 23 de enero de 2013. Además, el documento "Bit-depth scalability compatible to H.264/AVC-scalable extension" de Wu, Gau y Chen (doi:10.1016/j.jvcir.2008/06/003) divulga una escalabilidad combinada de profundidad de bits y espacial en la que el orden en que se implementa la escalabilidad se adapta en base al problema a abordar.

[0018] A continuación en el presente documento, se describen de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a los dibujos adjuntos. Sin embargo, la presente divulgación se puede realizar de muchas formas diferentes y no se debería interpretar que está limitada a ninguna

estructura o función específicas presentadas a lo largo de esta divulgación. En su lugar, estos aspectos se proporcionan de modo que esta divulgación sea exhaustiva y completa, y transmita en su totalidad el alcance de la divulgación a los expertos en la técnica. En base a las enseñanzas del presente documento, un experto en la materia debería apreciar que el alcance de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en el presente documento, ya estén implementados de forma independiente de, o en combinación con, cualquier otro aspecto de la presente divulgación. Por ejemplo, un aparato se puede implementar o un procedimiento se puede llevar a la práctica usando un número cualquiera de los aspectos expuestos en el presente documento. Además, el alcance de la presente divulgación pretende abarcar dicho aparato o procedimiento que se lleva a la práctica usando otra estructura, funcionalidad o estructura y funcionalidad de forma adicional o alternativa a los diversos aspectos de la presente divulgación expuestos en el presente documento. Se debería entender que cualquier aspecto divulgado en el presente documento se puede realizar mediante uno o más elementos de una reivindicación.

[0019] Aunque en el presente documento se describen unos aspectos en particular, muchas variantes y permutaciones de estos aspectos se hallan dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, no se pretende que el alcance de la divulgación esté limitado a unos beneficios, usos u objetivos en particular. En su lugar, se pretende que los aspectos de la divulgación sean ampliamente aplicables a diferentes tecnologías inalámbricas, configuraciones de sistema, redes y protocolos de transmisión, algunos de los cuales se ilustran a modo de ejemplo en las figuras y en la siguiente descripción de los aspectos preferentes. La descripción detallada y los dibujos son meramente ilustrativos de la divulgación, en lugar de limitativos, estando definido el alcance de la divulgación por las reivindicaciones adjuntas y equivalentes de las mismas.

[0020] Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", y así sucesivamente) no implican necesariamente que los elementos tienen un orden en particular. En su lugar, dichas palabras ordinales se usan meramente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

Sistema de codificación de vídeo

[0021] La **FIG. 1** es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo 10 que puede utilizar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación. Como se usa en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" se pueden referir genéricamente a la codificación de vídeo y a la descodificación de vídeo.

[0022] Como se muestra en la **FIG. 1**, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de escritorio, cuadernos digitales (por ejemplo, portátiles, etc.), ordenadores de tableta, descodificadores, teléfonos portátiles tales como los denominados teléfonos "inteligentes", las denominadas pizarras "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de automóviles o similares. En algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0023] El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender un medio de comunicación que permite al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El medio de comunicación puede comprender un medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base u otros equipos que facilitan la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0024] En otro ejemplo, el canal 16 puede corresponder a un medio de almacenamiento que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento por medio de acceso de disco o de acceso de tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria *flash* u otros medios de almacenamiento digital adecuados para

almacenar datos de vídeo codificados. En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena el vídeo codificado generado por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos o en otro dispositivo de almacenamiento intermedio por medio de transmisión continua o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para un sitio web, etc.), servidores FTP, dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos se pueden incluir canales inalámbricos (por ejemplo, conexiones wifi, etc.), conexiones alámbricas (por ejemplo, DSL, módem de cable, etc.) o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0025] Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo para admitir cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, por medio de Internet (por ejemplo, transmisión continua dinámica adaptativa por HTTP (DASH), etc.), codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0026] En el ejemplo de la **FIG. 1**, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo previamente capturados, una interfaz de transmisión de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes.

[0027] El codificador de vídeo 20 puede estar configurado para codificar los datos de vídeo capturados, precapturados o generados por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también se pueden almacenar en un medio de almacenamiento o en un servidor de archivos para un acceso posterior por el dispositivo de destino 14 para su descodificación y/o su reproducción.

[0028] En el ejemplo de la **FIG. 1**, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del canal 16. Los datos de vídeo codificados pueden incluir una variedad de elementos sintácticos generados por el codificador de vídeo 20, que representan los datos de vídeo. Los elementos sintácticos pueden describir características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). Dichos elementos sintácticos se pueden incluir en los datos de vídeo codificados, transmitir en un medio de comunicación, almacenar en un medio de almacenamiento o almacenar en un servidor de archivos.

[0029] El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también puede estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario. El dispositivo de visualización 32 puede comprender cualquiera de entre una variedad de dispositivos de visualización, tales como 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.

[0030] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en desarrollo, y se pueden adaptar a un modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, Codificación avanzada de vídeo (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación en particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

[0031] Aunque no se muestra en el ejemplo de la **FIG. 1**, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados en un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para ocuparse de la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden adaptarse al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0032] Otra vez, la **FIG. 1** es meramente un ejemplo y las técnicas de esta divulgación se pueden aplicar a unas configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se pueden recuperar de una memoria local, transmitir en continuo a través de una red, o similares. Un dispositivo de codificación puede codificar y almacenar datos en una memoria, y/o un dispositivo de descodificación puede recuperar y descodificar datos de una memoria. En muchos ejemplos, la codificación y la descodificación se realizan mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y descodifican datos de una memoria.

[0033] El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar cada uno como cualquiera de entre una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento no transitorio legible por ordenador adecuado, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Aunque el codificador de vídeo 20 y el descodificador de vídeo 30 que se muestran están implementados en dispositivos separados en el ejemplo de la **FIG. 1**, la presente divulgación no se limita a dicha configuración, sino que el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar implementados en el mismo dispositivo. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

[0034] Como se ha mencionado anteriormente de manera breve, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen se puede denominar "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.

[0035] Para generar el flujo de bits, el codificador de vídeo 20 puede realizar unas operaciones de codificación en cada imagen de los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen, conjuntos de parámetros de adaptación y otras estructuras sintácticas. Un conjunto de parámetros de secuencia (SPS) puede contener parámetros aplicables a cero o más secuencias de imágenes. Un conjunto de parámetros de imagen (PPS) puede contener parámetros aplicables a cero o más imágenes. Un conjunto de parámetros de adaptación (APS) puede contener parámetros aplicables a cero o más imágenes. Los parámetros de un APS pueden ser parámetros que es más probable que cambien que los parámetros de un PPS.

[0036] Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño. Un bloque de vídeo puede ser una matriz bidimensional de muestras. Cada uno de los bloques de vídeo está asociado a un bloque de árbol. En algunos casos, un bloque de árbol también se puede denominar unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de normas previas, tales como la H.264/AVC. Sin embargo, un bloque de árbol no está limitado necesariamente a un tamaño en particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar una división en árbol cuaternario para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados a unas CU, de ahí el nombre "bloques de árbol".

[0037] En algunos ejemplos, el codificador de vídeo 20 puede dividir una imagen en una pluralidad de sectores. Cada uno de los sectores puede incluir un número entero de CU. En algunos casos, un sector comprende un número entero de bloques de árbol. En otros casos, un límite de un sector puede estar dentro de un bloque de árbol.

[0038] Como parte de la realización de una operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada sector de la imagen. Cuando el codificador de vídeo 20 realiza una operación de codificación en un sector, el codificador de vídeo 20 puede generar datos codificados asociados al sector. Los datos codificados asociados al sector se pueden denominar "sector codificado".

[0039] Para generar un sector codificado, el codificador de vídeo 20 puede realizar unas operaciones de codificación en cada bloque de árbol de un sector. Cuando el codificador de vídeo 20 realiza una operación de codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

[0040] Cuando el codificador de vídeo 20 genera un sector codificado, el codificador de vídeo 20 puede realizar operaciones de codificación (es decir, codificar) en los bloques de árbol del sector de acuerdo con un orden de barrido. Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del sector en un orden dirigido de izquierda a derecha en la fila más alta de los bloques de árbol del sector, a continuación de izquierda a derecha en la siguiente fila inferior de los bloques de árbol, y así sucesivamente hasta que el codificador de vídeo 20 ha codificado cada uno de los bloques de árbol del sector.

[0041] Como resultado de la codificación de los bloques de árbol de acuerdo con el orden de barrido, aunque los bloques de árbol de encima y de la izquierda de un bloque de árbol dado se puedan haber codificado, los bloques de árbol de debajo y de la derecha del bloque de árbol dado aún no se habrán codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a la información generada por la codificación de los bloques de árbol de encima y de la izquierda del bloque de árbol dado cuando se codifica el bloque de árbol dado. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a la información generada por la codificación de los bloques de árbol de debajo y de la derecha del bloque de árbol dado cuando se codifica el bloque de árbol dado.

[0042] Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede realizar de forma recursiva una división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente. Una CU dividida puede ser una CU cuyo bloque de vídeo está dividido en bloques de vídeo asociados a otras CU. Una CU no dividida puede ser una CU cuyo bloque de vídeo no está dividido en bloques de vídeo asociados a otras CU.

[0043] Uno o más elementos sintácticos del flujo de bits pueden indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor.

[0044] El codificador de vídeo 20 puede realizar operaciones de codificación (es decir, codificar) en cada CU de un bloque de árbol de acuerdo con un orden de exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y a continuación una CU inferior derecha, en ese orden. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU dividida, el codificador de vídeo 20 puede codificar las CU asociadas a subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU asociada a un subbloque superior izquierdo, una CU asociada a un subbloque superior derecho, una CU asociada a un subbloque inferior izquierdo y a continuación una CU asociada a un subbloque inferior derecho, en ese orden.

[0045] Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de exploración en z, las CU de encima, de encima y a la izquierda, de encima y a la derecha, de la izquierda y de debajo y a la izquierda de una CU dada se pueden haber codificado. Las CU de debajo y a la derecha de la CU determinada todavía no se han codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a información generada por la codificación de algunas CU vecinas de la CU dada cuando se codifica la CU dada. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a información generada por la codificación de otras CU vecinas de la CU dada cuando se codifica la CU dada.

[0046] Cuando el codificador de vídeo 20 codifica una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada a un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar el bloque de vídeo predicho para una PU.

- 5 [0047] Cuando el codificador de vídeo 20 usa intrapredicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU en base a unas muestras descodificadas de la imagen asociada a la PU. Si el codificador de vídeo 20 usa intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Cuando el codificador de vídeo 20 usa interpredicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU en base a unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. Si el codificador de vídeo 20 usa interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.
- 10 [0048] Además, cuando el codificador de vídeo 20 usa interpredicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de vídeo dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada a la PU. En algunos casos, un bloque de referencia de una PU también se puede denominar "muestra de referencia" de la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU en base a los bloques de referencia de la PU.
- 15 [0049] Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU en base a los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.
- 20 [0050] Además, como parte de la realización de una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar una división recursiva en árbol cuaternario en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de vídeo residuales) asociados a las unidades de transformada (TU) de la CU. Cada TU de una CU puede estar asociada a un bloque de vídeo residual diferente.
- 25 [0051] El codificador de vídeo 20 puede aplicar una o más transformadas a los bloques de vídeo residuales asociados a las TU para generar bloques de coeficientes de transformada (por ejemplo, bloques de coeficientes de transformada) asociados a las TU. Conceptualmente, un bloque de coeficientes de transformada puede ser una matriz bidimensional (2D) de coeficientes de transformada.
- 30 [0052] Después de generar un bloque de coeficientes de transformada, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits se puede redondear a la baja hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m .
- 35 [0053] El codificador de vídeo 20 puede asociar cada CU a un valor de parámetro de cuantificación (QP). El valor de QP asociado a una CU puede determinar cómo cuantifica el codificador de vídeo 20 los bloques de coeficientes de transformada asociados a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformada asociados a una CU ajustando el valor de QP asociado a la CU.
- 40 [0054] Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes de transformada, el codificador de vídeo 20 puede generar conjuntos de elementos sintácticos que representan los coeficientes de transformada en el bloque de coeficientes de transformada cuantificados. El codificador de vídeo 20 puede aplicar operaciones de codificación entrópica, tales como operaciones de codificación aritmética binaria adaptativa al contexto (CABAC), a algunos de estos elementos sintácticos. También se podrían usar otras técnicas de codificación entrópica, tales como la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación entrópica de división de intervalo de probabilidad (PIPE) u otra codificación aritmética binaria.
- 45 [0055] El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contiene una indicación de un tipo de datos de la unidad NAL y bytes que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un sector codificado, información de mejora complementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos de una unidad NAL pueden incluir diversas estructuras sintácticas.
- 50 [0056] El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. El flujo de bits puede incluir una representación codificada de los datos de vídeo codificados por el codificador de vídeo 20. Cuando el descodificador de vídeo 30 recibe el flujo de bits, el descodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el descodificador de vídeo 30 realiza la operación de
- 55
- 60
- 65

análisis sintáctico, el descodificador de vídeo 30 puede extraer elementos sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a los elementos sintácticos extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo en base a los elementos sintácticos puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos sintácticos.

[0057] Después de que el descodificador de vídeo 30 extraiga los elementos sintácticos asociados a una CU, el descodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU en base a los elementos sintácticos. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa de unos bloques de coeficientes de transformada asociados a las TU de la CU. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de vídeo residuales asociados a las TU de la CU. Después de generar los bloques de vídeo predichos y reconstruir los bloques de vídeo residuales, el descodificador de vídeo 30 puede reconstruir el bloque de vídeo de la CU en base a los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede reconstruir los bloques de vídeo de las CU en base a los elementos sintácticos del flujo de bits.

Codificador de vídeo

[0058] La **FIG. 2A** es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación. El codificador de vídeo 20 puede estar configurado para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 puede estar configurado para realizar cualquiera o la totalidad de las técnicas de esta divulgación. Como ejemplo, la unidad de procesamiento de predicción 100 puede estar configurada para realizar cualquiera o la totalidad de las técnicas descritas en esta divulgación. En otro modo de realización, el codificador de vídeo 20 incluye una unidad de predicción entre capas 128 opcional que está configurada para realizar cualquiera o la totalidad de las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar mediante la unidad de procesamiento de predicción 100 (por ejemplo, la unidad de interpredicción 121 y/o la unidad de intrapredicción 126), en cuyo caso la unidad de predicción entre capas 128 se puede omitir. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación se pueden compartir entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, de forma adicional o alternativa, un procesador (no mostrado) puede estar configurado para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación.

[0059] Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **FIG. 2A** es para un códec de una sola capa. Sin embargo, como se describirá en mayor detalle con respecto a la **FIG. 2B**, una parte o la totalidad del codificador de vídeo 20 se puede duplicar para procesar un códec multicapa.

[0060] El codificador de vídeo 20 puede realizar una intra- y una intercodificación de bloques de vídeo dentro de sectores de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) se puede referir a cualquiera de varios modos de codificación basados en el espacio. Los modos inter, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), se pueden referir a cualquiera de varios modos de codificación basados en el tiempo.

[0061] En el ejemplo de la **FIG. 2A**, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, una memoria intermedia de imágenes descodificadas 114 y una unidad de codificación entrópica 116. La unidad de procesamiento de predicción 100 incluye una unidad de interpredicción 121, una unidad de estimación de movimiento 122, una unidad de compensación de movimiento 124, una unidad de intrapredicción 126 y una unidad de predicción entre capas 128. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden estar altamente integradas, aunque en el ejemplo de la **FIG. 2A** están representadas de forma separada con propósitos explicativos.

[0062] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo desde diversas fuentes. Por ejemplo, el codificador de vídeo 20 puede recibir los datos de vídeo desde la fuente de vídeo 18 (**FIG. 1**) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte de la realización de la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada sector de la imagen. Como parte de la realización de

una operación de codificación en un sector, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el sector.

[0063] Como parte de la realización de una operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede realizar una división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente.

[0064] Los tamaños de los bloques de vídeo asociados a las CU pueden variar desde muestras de 8x8 hasta el tamaño del bloque de árbol, con un máximo de muestras de 64x64 o mayor. En esta divulgación, "NxN" y "N por N" se pueden usar de manera intercambiable para hacer referencia a las dimensiones de muestras de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ($y = 16$) y dieciséis muestras en una dirección horizontal ($x = 16$). Asimismo, un bloque NxN en general tiene N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

[0065] Además, como parte de la realización de la operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede generar una estructura de datos jerárquica en árbol cuaternario para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuaternario. Si la unidad de procesamiento de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos hijo en la estructura de datos en árbol cuaternario. Cada uno de los nodos hijo corresponde a una CU asociada a uno de los subbloques. Si la unidad de procesamiento de predicción 100 divide uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada al subbloque puede tener cuatro nodos hijo, cada uno de los cuales corresponde a una CU asociada a uno de los subsubbloques.

[0066] Cada nodo de la estructura de datos en árbol cuaternario puede contener datos sintácticos (por ejemplo, elementos sintácticos) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está dividido (es decir, separado) en cuatro subbloques. Los elementos sintácticos para una CU se pueden definir de manera recursiva y pueden depender de si el bloque de vídeo de la CU está separado en subbloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuaternario. Un bloque de árbol codificado puede incluir datos en base a la estructura de datos en árbol cuaternario para un bloque de árbol correspondiente.

[0067] El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU no dividida, el codificador de vídeo 20 genera datos que representan una representación codificada de la CU no dividida.

[0068] Como parte de la realización de una operación de codificación en una CU, la unidad de procesamiento de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir diversos tamaños de PU. Suponiendo que el tamaño de una CU en particular sea $2N \times 2N$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ e interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ o similares. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden admitir la división asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En algunos ejemplos, la unidad de procesamiento de predicción 100 puede realizar una división geométrica para dividir el bloque de vídeo de una CU entre las PU de la CU a lo largo de un límite que no forma ángulos rectos con los lados del bloque de vídeo de la CU.

[0069] La unidad de interpredicción 121 puede realizar la interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar la interpredicción en una PU, la unidad de estimación de movimiento 122 puede generar información de movimiento para la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho para la PU en base a la información de movimiento y las muestras decodificadas de imágenes distintas a la imagen asociada a la CU (por ejemplo, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar bloque de vídeo interpredicho.

[0070] Los sectores pueden ser sectores I, sectores P o sectores B. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un sector I, un sector P o un sector B. En un sector I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un sector I, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 no realizan la interpredicción en la PU.

- 5 **[0071]** Si la PU está en un sector P, la imagen que contiene la PU está asociada a una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia de la lista 0 contiene muestras que se pueden usar para la interpredicción de otras imágenes. Cuando la unidad de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU de un sector P, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de la lista 0 un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que se corresponde más estrechamente con las muestras del bloque de vídeo de la PU. La unidad de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras de una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras de una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU mediante una suma de diferencias absolutas (SAD), una suma de diferencias cuadráticas (SSD) u otras métricas de diferencia.
- 15 **[0072]** Después de identificar un bloque de referencia de una PU en un sector P, la unidad de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En diversos ejemplos, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con una precisión de un cuarto de muestra, una precisión de un octavo de muestra u otra precisión de muestra fraccionaria. En el caso de la precisión de muestra fraccionaria, los valores de bloque de referencia se pueden interpolar a partir de valores de muestra de posición enteros en la imagen de referencia. La unidad de estimación de movimiento 122 puede facilitar el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU en base al bloque de referencia identificado por la información de movimiento de la PU.
- 30 **[0073]** Si la PU está en un sector B, la imagen que contiene la PU puede estar asociada a dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1". En algunos ejemplos, una imagen que contiene un sector B puede estar asociada a una combinación de listas que es una combinación de la lista 0 y la lista 1.
- 35 **[0074]** Además, si la PU está en un sector B, la unidad de estimación de movimiento 122 puede realizar una predicción unidireccional o una predicción bidireccional para la PU. Cuando la unidad de estimación de movimiento 122 realiza la predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de la lista 0 o la lista 1 un bloque de referencia para la PU. La unidad de estimación de movimiento 122 a continuación puede generar un índice de referencia que indica la imagen de referencia de la lista 0 o la lista 1 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. La unidad de estimación de movimiento 122 puede facilitar el índice de referencia, un indicador de dirección de predicción y el vector de movimiento como la información de movimiento de la PU. El indicador de dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia de la lista 0 o la lista 1. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU en base al bloque de referencia indicado por la información de movimiento de la PU.
- 45 **[0075]** Cuando la unidad de estimación de movimiento 122 realiza la predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de la lista 0 un bloque de referencia para la PU y también puede buscar en las imágenes de referencia de la lista 1 otro bloque de referencia para la PU. A continuación, la unidad de estimación de movimiento 122 puede generar índices de referencia que indican las imágenes de referencia de la lista 0 y la lista 1 que contienen los bloques de referencia y los vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La unidad de estimación de movimiento 122 puede facilitar los índices de referencia y los vectores de movimiento de la PU como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU en base a los bloques de referencia indicados por la información de movimiento de la PU.
- 55 **[0076]** En algunos casos, la unidad de estimación de movimiento 122 no facilita un conjunto completo de información de movimiento para una PU a la unidad de codificación entrópica 116. En su lugar, la unidad de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU vecina. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura sintáctica asociada a la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU vecina. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura sintáctica asociada a la PU, una PU vecina y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU vecina indicada. El descodificador de vídeo 30 puede usar el vector de movimiento de la PU vecina indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Con

referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

5 **[0077]** Como se analiza en mayor detalle a continuación con referencia a las **FIGS. 8 y 9**, la unidad de procesamiento de predicción 100 puede estar configurada para codificar (por ejemplo, codificar o decodificar) la PU (o cualquier otro bloque o unidad de vídeo de capa de mejora) realizando los procedimientos ilustrados en las **FIGS. 8 y 9**. Por ejemplo, la unidad de interpredicción 121 (por ejemplo, por medio de la unidad de estimación de movimiento 122 y/o la unidad de compensación de movimiento 124), la unidad de intrapredicción 126, o la unidad de predicción entre capas 128 pueden estar configuradas para realizar los procedimientos ilustrados en las **FIGS. 8 y 9** ya sea juntas o por separado.

15 **[0078]** Como parte de la realización de una operación de codificación en una CU, la unidad de intrapredicción 126 puede realizar una intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando la unidad de intrapredicción 126 realiza una intrapredicción en una PU, la unidad de intrapredicción 126 puede generar datos de predicción para la PU en base a unas muestras decodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y diversos elementos sintácticos. La unidad de intrapredicción 126 puede realizar la intrapredicción en PU de sectores I, sectores P y sectores B.

20 **[0079]** Para realizar una intrapredicción en una PU, la unidad de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando la unidad de intrapredicción 126 usa un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, la unidad de intrapredicción 126 puede extender muestras de bloques de vídeo de PU vecinas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociados al modo de intrapredicción. Las PU vecinas pueden estar encima, encima y a la derecha, encima y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo para las PU, CU y bloques de árbol. La unidad de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

30 **[0080]** La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU de entre los datos de predicción generados por la unidad de compensación de movimiento 124 para la PU o los datos de predicción generados por la unidad de intrapredicción 126 para la PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción para la PU en base a las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

35 **[0081]** Si la unidad de procesamiento de predicción 100 selecciona datos de predicción generados por la unidad de intrapredicción 126, la unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción que se ha usado para generar los datos de predicción para las PU, por ejemplo, el modo de intrapredicción seleccionado. La unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción seleccionado de diversas maneras. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. Por tanto, la unidad de procesamiento de predicción 100 puede generar un elemento sintáctico para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU vecina.

40 **[0082]** Como se analiza anteriormente, el codificador de vídeo 20 puede incluir una unidad de predicción entre capas 128. La unidad de predicción entre capas 128 está configurada para predecir un bloque actual (por ejemplo, un bloque actual de la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 128 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando de este modo la eficacia de codificación y reduciendo los requisitos computacionales de recursos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas usa la reconstrucción de bloques colocados en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. A continuación, se analiza en mayor detalle cada uno de los sistemas de predicción entre capas.

50 **[0083]** Después de que la unidad de procesamiento de predicción 100 seleccione los datos de predicción para las PU de una CU, la unidad de generación residual 102 puede generar datos residuales para la CU restando (por ejemplo, indicado por el signo menos) los bloques de vídeo predichos de las PU de la CU a partir del bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales 2D que corresponden a diferentes componentes de muestra de las muestras del bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a unas diferencias entre unos componentes de luminancia de unas muestras de los bloques de vídeo predichos de las PU de la CU y unos componentes de

luminancia de unas muestras del bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre unos componentes de crominancia de unas muestras de los bloques de vídeo predichos de las PU de la CU y los componentes de crominancia de las muestras del bloque de vídeo original de la CU.

[0084] La unidad de procesamiento de predicción 100 puede realizar una división en árbol cuaternario para dividir los bloques de vídeo residuales de una CU en subbloques. Cada bloque de vídeo residual no dividido puede estar asociado a una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados a las TU de una CU pueden estar basados o no en los tamaños y las posiciones de unos bloques de vídeo asociados a las PU de la CU. Una estructura en árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0085] La unidad de procesamiento de transformada 104 puede generar uno o más bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a un bloque de vídeo residual asociado a la TU. Cada uno de los bloques de coeficientes de transformada puede ser una matriz 2D de coeficientes de transformada. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas al bloque de vídeo residual asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de vídeo residual asociado a una TU.

[0086] Después de que la unidad de procesamiento de transformada 104 genere un bloque de coeficientes de transformada asociado a una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformada del bloque de coeficientes de transformada. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformada asociado a una TU de una CU en base a un valor de QP asociado a la CU.

[0087] El codificador de vídeo 20 puede asociar un valor de QP a una CU de diversas maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado a la CU. En el análisis de velocidad-distorsión, el codificador de vídeo 20 puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación múltiples veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP a la CU cuando el codificador de vídeo 20 genera diferentes representaciones codificadas del bloque de árbol. El codificador de vídeo 20 puede señalar que un valor de QP dado está asociado a la CU cuando el valor de QP dado está asociado a la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

[0088] La unidad de cuantificación inversa 108 y la unidad de transformada inversa 110 pueden aplicar una cuantificación inversa y unas transformadas inversas al bloque de coeficientes de transformada, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformada. La unidad de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a unas muestras correspondientes de uno o más bloques de vídeo predichos generados por la unidad de procesamiento de predicción 100 para generar un bloque de vídeo reconstruido asociado a una TU. Reconstruyendo bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

[0089] Después de que la unidad de reconstrucción 112 reconstruya el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de eliminación de bloques para reducir los artefactos de bloque en el bloque de vídeo asociado a la CU. Después de realizar la una o más operaciones de eliminación de bloques, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en una memoria intermedia de imágenes descodificadas 114. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden usar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar una interpredicción en las PU de las imágenes posteriores. Además, la unidad de intrapredicción 126 puede usar bloques de vídeo reconstruidos de la memoria intermedia de imágenes descodificadas 114 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

[0090] La unidad de codificación entrópica 116 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación entrópica 116 puede recibir bloques de coeficientes de transformada desde la unidad de cuantificación 106 y puede recibir elementos sintácticos desde la unidad de procesamiento de predicción 100. Cuando la unidad de codificación entrópica 116 recibe los datos, la unidad de codificación entrópica 116 puede realizar una o más operaciones de codificación entrópica para generar datos codificados entrópicamente. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud de variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación entrópica de división de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación entrópica en los datos. La unidad de codificación entrópica 116 puede facilitar un flujo de bits que incluye los datos codificados entrópicamente.

[0091] Como parte de la realización de una operación de codificación entrópica en unos datos, la unidad de codificación entrópica 116 puede seleccionar un modelo de contexto. Si la unidad de codificación entrópica 116 está realizando una operación CABAC, el modelo de contexto puede indicar unas estimaciones de unas probabilidades de que unos bins en particular tengan unos valores en particular. En el contexto de CABAC, el término "bin" se usa para hacer referencia a un bit de una versión binarizada de un elemento sintáctico.

Codificador de vídeo multicapa

[0092] La FIG. 2B es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo multicapa 21 que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación. El codificador de vídeo 21 puede estar configurado para procesar tramas de vídeo multicapa, tales como para codificación SHVC y multivista. Además, el codificador de vídeo 21 puede estar configurado para realizar una cualquiera o la totalidad de las técnicas de esta divulgación.

[0093] El codificador de vídeo 21 incluye un codificador de vídeo 20A y un codificador de vídeo 20B, cada uno de los cuales puede estar configurado como el codificador de vídeo 20 y puede realizar las funciones descritas anteriormente con respecto al codificador de vídeo 20. Además, como se indica mediante el reuso de números de referencia, los codificadores de vídeo 20A y 20B pueden incluir al menos algunos de los sistemas y subsistemas como el codificador de vídeo 20. Aunque el codificador de vídeo 21 ilustrado incluye dos codificadores de vídeo 20A y 20B, esto no limita al codificador de vídeo 21, sino que este puede incluir cualquier número de capas de codificador de vídeo 20. En algunos modos de realización, el codificador de vídeo 21 puede incluir un codificador de vídeo 20 para cada imagen o trama de una unidad de acceso. Por ejemplo, un codificador de vídeo que incluye cinco capas de codificador puede procesar o codificar una unidad de acceso que incluye cinco imágenes. En algunos modos de realización, el codificador de vídeo 21 puede incluir más capas de codificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas del codificador de vídeo pueden estar inactivas cuando se procesan algunas unidades de acceso.

[0094] Además de los codificadores de vídeo 20A y 20B, el codificador de vídeo 21 puede incluir una unidad de remuestreo 90. La unidad de remuestreo 90 puede, en algunos casos, realizar un sobremuestreo de una capa base de una trama de vídeo recibida para, por ejemplo, crear una capa de mejora. La unidad de remuestreo 90 puede realizar un sobremuestreo de información en particular asociada a la capa base recibida de una trama, pero no de otra información. Por ejemplo, la unidad de remuestreo 90 puede realizar sobremuestreo del tamaño espacial o el número de píxeles de la capa base, pero el número de sectores o el recuento de orden de imágenes puede permanecer constante. En algunos casos, la unidad de remuestreo 90 puede no procesar el vídeo recibido y/o puede ser opcional. Por ejemplo, en algunos casos, la unidad de procesamiento de predicción 100 puede realizar un sobremuestreo. En algunos modos de realización, la unidad de remuestreo 90 está configurada para sobremuestrear una capa y reorganizar, redefinir, modificar o ajustar uno o más sectores para cumplir con un conjunto de reglas de límite de sector y/o reglas de barrido. Aunque se describe principalmente como un sobremuestreo de una capa base, o de una capa inferior en una unidad de acceso, en algunos casos, la unidad de remuestreo 90 puede submuestrear una capa. Por ejemplo, si durante una transmisión en continuo de un vídeo el ancho de banda se reduce, en lugar de sobremuestrear una trama, esta se puede submuestrear.

[0095] La unidad de remuestreo 90 puede estar configurada para recibir una imagen o una trama (o información de imagen asociada a la imagen) desde la memoria intermedia de imágenes descodificadas 114 del codificador de capa inferior (por ejemplo, el codificador de vídeo 20A) y para sobremuestrear la imagen (o la información de imagen recibida). Esta imagen sobremuestreada se puede proporcionar a continuación a la unidad de procesamiento de predicción 100 de un codificador de capa superior (por ejemplo, el codificador de vídeo 20B) configurado para codificar una imagen en la misma unidad de acceso que el codificador de capa inferior. En algunos casos, el codificador de capa superior es una capa eliminada del codificador de capa inferior. En otros casos, puede haber uno o más codificadores de capa superior entre el codificador de vídeo de capa 0 y el codificador de capa 1 de la FIG. 2B.

[0096] En algunos casos, la unidad de remuestreo 90 se puede omitir o excluir. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A se puede proporcionar directamente, o al menos sin proporcionarla a la unidad de remuestreo 90 y a la unidad de procesamiento de predicción 100 del codificador de vídeo 20B. Por ejemplo, si los datos de vídeo proporcionados al codificador de vídeo 20B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al codificador de vídeo 20B sin ningún remuestreo.

[0097] En algunos modos de realización, el codificador de vídeo 21 realiza un submuestreo de los datos de vídeo que se van a proporcionar al codificador de capa inferior usando la unidad de submuestreo 94 antes de proporcionar los datos de vídeo al codificador de vídeo 20A. De forma alternativa, la unidad de submuestreo 94 puede ser una unidad de remuestreo 90 capaz de sobremuestrear o submuestrear los datos de vídeo. En otros modos de realización más, la unidad de submuestreo 94 se puede omitir.

[0098] Como se ilustra en la **FIG. 2B**, el codificador de vídeo 21 puede incluir además un multiplexor 98 o mux. El mux 98 puede facilitar un flujo de bits combinado desde el codificador de vídeo 21. El flujo de bits combinado se puede crear tomando un flujo de bits de cada uno de los codificadores de vídeo 20A y 20B y alternando el flujo de bits que se facilita en un momento dado. Mientras que en algunos casos los bits de los dos flujos de bits (o más en el caso de más de dos capas de codificador de vídeo) se pueden alternar de uno en uno, en muchos casos los flujos de bits se combinan de manera diferente. Por ejemplo, el flujo de bits facilitado se puede crear alternando el flujo de bits seleccionado de bloque en bloque. En otro ejemplo, el flujo de bits facilitado se puede crear facilitando una relación de bloques no 1:1 de cada uno de los codificadores de vídeo 20A y 20B. Por ejemplo, se pueden facilitar dos bloques desde el codificador de vídeo 20B para cada bloque facilitado desde el codificador de vídeo 20A. En algunos modos de realización, el flujo de salida del mux 98 se puede preprogramar. En otros modos de realización, el mux 98 puede combinar los flujos de bits de los codificadores de vídeo 20A, 20B en base a una señal de control recibida desde un sistema externo al codificador de vídeo 21, tal como desde un procesador en un dispositivo de origen 12. La señal de control se puede generar en base a la resolución o velocidad de bits de un vídeo desde la fuente de vídeo 18, en base a un ancho de banda del canal 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una resolución facilitada deseada desde el codificador de vídeo 21.

Descodificador de vídeo

[0099] La **FIG. 3A** es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación. El descodificador de vídeo 30 puede estar configurado para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el descodificador de vídeo 30 puede estar configurado para realizar una cualquiera o la totalidad de las técnicas de esta divulgación. Como ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164 pueden estar configuradas para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación. En un modo de realización, el descodificador de vídeo 30 puede incluir opcionalmente una unidad de predicción entre capas 166 que está configurada para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar mediante la unidad de procesamiento de predicción 152 (por ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164), en cuyo caso la unidad de predicción entre capas 166 se puede omitir. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación se pueden compartir entre los diversos componentes del descodificador de vídeo 30. En algunos ejemplos, de forma adicional o alternativa, un procesador (no mostrado) puede estar configurado para realizar una cualquiera o la totalidad de las técnicas descritas en esta divulgación.

[0100] Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **FIG. 3A** es para un códec de una sola capa. Sin embargo, como se describe en mayor detalle con respecto a la **FIG. 3B**, una parte o la totalidad del descodificador de vídeo 30 se puede duplicar para el procesamiento de un códec multicapa.

[0101] En el ejemplo de la **FIG. 3A**, el descodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del descodificador de vídeo 30 incluyen una unidad de descodificación entrópica 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y una memoria intermedia de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 162, una unidad de intrapredicción 164 y una unidad de predicción entre capas 166. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la **FIG. 2A**. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0102] El descodificador de vídeo 30 puede recibir un flujo de bits que comprende datos de vídeo codificados. El flujo de bits puede incluir una pluralidad de elementos sintácticos. Cuando el descodificador de vídeo 30 recibe el flujo de bits, la unidad de descodificación entrópica 150 puede realizar una operación de análisis sintáctico en el flujo de bits. Como resultado de la realización de la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación entrópica 150 puede extraer elementos sintácticos del flujo de bits. Como parte de la realización de la operación de análisis sintáctico, la unidad de descodificación entrópica 150 puede descodificar entrópicamente elementos sintácticos codificados entrópicamente en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden realizar una operación de reconstrucción que genera datos de vídeo descodificados en base a los elementos sintácticos extraídos del flujo de bits.

[0103] Como se analiza anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de vídeo, unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades NAL SEI, y así sucesivamente. Como parte de la realización de la operación de análisis sintáctico en el flujo de bits, la unidad

de descodificación entrópica 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican entrópicamente conjuntos de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades NAL SEI, y así sucesivamente.

5

[0104] Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de sectores codificados. Como parte de la realización de la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación entrópica 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican entrópicamente sectores codificados de las unidades NAL de sectores codificados. Cada uno de los sectores codificados puede incluir una cabecera de sector y datos de sector. La cabecera de sector puede contener elementos sintácticos pertenecientes a un sector. Los elementos sintácticos de la cabecera de sector pueden incluir un elemento sintáctico que identifica un conjunto de parámetros de imagen asociado a una imagen que contiene el sector. La unidad de descodificación entrópica 150 puede realizar operaciones de descodificación entrópica, tales como operaciones de descodificación CABAC, en los elementos sintácticos de la cabecera de sector codificado para recuperar la cabecera del sector.

10

15

[0105] Como parte de la extracción de los datos de sector de unas unidades NAL de sectores codificados, la unidad de descodificación entrópica 150 puede realizar operaciones de análisis sintáctico que extraen los elementos sintácticos de las CU codificadas en los datos del sector. Los elementos sintácticos extraídos pueden incluir elementos sintácticos asociados a bloques de coeficientes de transformada. La unidad de descodificación entrópica 150 a continuación puede realizar operaciones de descodificación CABAC en algunos de los elementos sintácticos.

20

[0106] Después de que la unidad de descodificación entrópica 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Al realizar la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado a la CU.

25

[0107] Como parte de la realización de una operación de reconstrucción en una TU, la unidad de cuantificación inversa 154 puede cuantificar inversamente, es decir, descuantificar, un bloque de coeficientes de transformada asociado a la TU. La unidad de cuantificación inversa 154 puede cuantificar inversamente el bloque de coeficientes de transformada de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por la norma de descodificación H.264. La unidad de cuantificación inversa 154 puede usar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformada para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique.

30

35

[0108] Después de que la unidad de cuantificación inversa 154 cuantifique inversamente un bloque de coeficientes de transformada, la unidad de transformada inversa 156 puede generar un bloque de vídeo residual para la TU asociada al bloque de coeficientes de transformada. La unidad de transformada inversa 156 puede aplicar una transformada inversa al bloque de coeficientes de transformada para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformada inversa 156 puede aplicar una DCT inversa, una transformada de entero inversa, una transformada de Karhunen-Loeve (KLT) inversa, una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes de transformada. En algunos ejemplos, la unidad de transformada inversa 156 puede determinar una transformada inversa que se va a aplicar al bloque de coeficientes de transformada en base a la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformada inversa 156 puede determinar la transformada inversa en base a una transformada señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado al bloque de coeficientes de transformada. En otros ejemplos, la unidad de transformada inversa 156 puede deducir la transformada inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, la unidad de transformada inversa 156 puede aplicar una transformada inversa en cascada.

40

45

50

[0109] En algunos ejemplos, la unidad de compensación de movimiento 162 puede mejorar el bloque de vídeo predicho de una PU realizando una interpolación en base a unos filtros de interpolación. Los identificadores para los filtros de interpolación que se van a usar para la compensación de movimiento con una precisión de submuestra pueden estar incluidos en los elementos sintácticos. La unidad de compensación de movimiento 162 puede usar los mismos filtros de interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras de subenteros de un bloque de referencia. La unidad de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información sintáctica recibida y usar los filtros de interpolación para generar el bloque de vídeo predicho.

55

60

[0110] Como se analiza en mayor detalle más adelante con referencia a las **FIGS. 8 y 9**, la unidad de procesamiento de predicción 152 puede codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa, bloque o unidad de vídeo de capa de mejora) realizando los procedimientos ilustrados en las **FIGS. 8 y 9**.

65

Por ejemplo, la unidad de compensación de movimiento 162, la unidad de intrapredicción 164 o la unidad de predicción entre capas 166 pueden estar configuradas para realizar los procedimientos ilustrados en las **FIGS. 8 y 9**, ya sea juntas o por separado.

5 **[0111]** Si se codifica una PU usando intrapredicción, la unidad de intrapredicción 164 puede realizar la intrapredicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, la unidad de intrapredicción 164 puede determinar un modo de intrapredicción para la PU en base a unos elementos sintácticos del flujo de bits. El flujo de bits puede incluir elementos sintácticos que la unidad de intrapredicción 164 puede usar para determinar el modo de intrapredicción de la PU.

10 **[0112]** En algunos casos, los elementos sintácticos pueden indicar que la unidad de intrapredicción 164 va a usar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. Por consiguiente, en este ejemplo, el flujo de bits puede incluir un pequeño elemento sintáctico que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU vecina. A continuación, la unidad de intrapredicción 164 puede usar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU en base a los bloques de vídeo de las PU vecinas espacialmente.

20 **[0113]** Como se analiza anteriormente, el descodificador de vídeo 30 también puede incluir una unidad de predicción entre capas 166. La unidad de predicción entre capas 166 está configurada para predecir un bloque actual (por ejemplo, un bloque actual de la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 166 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando de este modo la eficacia de codificación y reduciendo los requisitos computacionales de recursos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas usa la reconstrucción de bloques colocalizados en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. A continuación, se analiza en mayor detalle cada uno de los sistemas de predicción entre capas.

35 **[0114]** La unidad de reconstrucción 158 puede usar los bloques de vídeo residuales asociados a las TU de una CU y los bloques de vídeo predichos de las PU de la CU, por ejemplo, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de vídeo de la CU. Por tanto, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual en base a unos elementos sintácticos del flujo de bits y puede generar un bloque de vídeo en base al bloque de vídeo predicho y el bloque de vídeo residual.

40 **[0115]** Después de que la unidad de reconstrucción 158 reconstruya el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de eliminación de bloques para reducir los artefactos de bloque asociados a la CU. Después de que la unidad de filtro 159 realice una operación de eliminación de bloques para reducir los artefactos de bloque asociados a la CU, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la **FIG. 1**. Por ejemplo, el descodificador de vídeo 30 puede realizar, en base a los bloques de vídeo de la memoria intermedia de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

Descodificador multicapa

55 **[0116]** La **FIG. 3B** es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo multicapa 31 que puede implementar unas técnicas de acuerdo con unos aspectos descritos en esta divulgación. El descodificador de vídeo 31 puede estar configurado para procesar tramas de vídeo multicapa, tal como para codificación SHVC y multivista. Además, el descodificador de vídeo 31 puede estar configurado para realizar una cualquiera o la totalidad de las técnicas de esta divulgación.

60 **[0117]** El descodificador de vídeo 31 incluye un descodificador de vídeo 30A y un descodificador de vídeo 30B, cada uno de los cuales puede estar configurado como el descodificador de vídeo 30 y puede realizar las funciones descritas anteriormente con respecto al descodificador de vídeo 30. Además, como se indica mediante el reuso de números de referencia, los descodificadores de vídeo 30A y 30B pueden incluir al menos algunos de los sistemas y subsistemas como el descodificador de vídeo 30. Aunque el descodificador de vídeo 31 que se ilustra incluye dos descodificadores de vídeo 30A y 30B, el descodificador de vídeo 31 no está limitado de ese modo y puede incluir cualquier número de capas de descodificador de vídeo 30. En algunos modos de realización, el

descodificador de vídeo 31 puede incluir un descodificador de vídeo 30 para cada imagen o trama de una unidad de acceso. Por ejemplo, un descodificador de vídeo que incluye cinco capas de descodificador puede procesar o descodificar una unidad de acceso que incluye cinco imágenes. En algunos modos de realización, el descodificador de vídeo 31 puede incluir más capas de descodificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas de descodificador de vídeo pueden estar inactivas cuando se procesan ciertas unidades de acceso.

[0118] Además de los descodificadores de vídeo 30A y 30B, el descodificador de vídeo 31 puede incluir una unidad de sobremuestreo 92. En algunos modos de realización, la unidad de sobremuestreo 92 puede sobremuestrear una capa base de una trama de vídeo recibida para crear una capa mejorada que se va a añadir a la lista de imágenes de referencia para la trama o la unidad de acceso. Esta capa mejorada se puede almacenar en la memoria intermedia de imágenes descodificadas 160. En algunos modos de realización, la unidad de sobremuestreo 92 puede incluir algunos o todos los modos de realización descritos con respecto a la unidad de remuestreo 90 de la **FIG. 2A**. En algunos modos de realización, la unidad de sobremuestreo 92 está configurada para sobremuestrear una capa y reorganizar, redefinir, modificar o ajustar uno o más sectores para cumplir con un conjunto de reglas de límite de sector y/o reglas de barrido. En algunos casos, la unidad de sobremuestreo 92 puede ser una unidad de remuestreo configurada para sobremuestrear y/o submuestrear una capa de una trama de vídeo recibida.

[0119] La unidad de sobremuestreo 92 puede estar configurada para recibir una imagen o trama (o información de imagen asociada a la imagen) desde la memoria intermedia de imágenes descodificadas 160 del descodificador de capa inferior (por ejemplo, el descodificador de vídeo 30A) y para sobremuestrear la imagen (o la información de imagen recibida). Esta imagen sobremuestreada se puede proporcionar a continuación a la unidad de procesamiento de predicción 152 de un descodificador de capa superior (por ejemplo, el descodificador de vídeo 30B) configurado para descodificar una imagen en la misma unidad de acceso que el descodificador de capa inferior. En algunos casos, el descodificador de capa superior es una capa eliminada del descodificador de capa inferior. En otros casos, puede haber uno o más descodificadores de capa superior entre el descodificador de capa 0 y el descodificador de capa 1 de la **FIG. 3B**.

[0120] En algunos casos, la unidad de sobremuestreo 92 se puede omitir o excluir. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A se puede proporcionar directamente, o al menos sin proporcionarla a la unidad de sobremuestreo 92, a la unidad de procesamiento de predicción 152 del descodificador de vídeo 30B. Por ejemplo, si los datos de vídeo proporcionados al descodificador de vídeo 30B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al descodificador de vídeo 30B sin sobremuestreo. Además, en algunos modos de realización, la unidad de sobremuestreo 92 puede ser una unidad de remuestreo 90 configurada para sobremuestrear o submuestrear una imagen de referencia recibida desde la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A.

[0121] Tal como se ilustra en la **FIG. 3B**, el descodificador de vídeo 31 puede incluir además un demultiplexor 99 o demux. El demux 99 puede dividir un flujo de bits de vídeo codificado en múltiples flujos de bits, proporcionándose cada flujo de bits facilitado por el demux 99 a un descodificador de vídeo diferente 30A y 30B. Los múltiples flujos de bits se pueden crear al recibir un flujo de bits, y cada uno de los descodificadores de vídeo 30A y 30B recibe una parte del flujo de bits en un momento dado. Mientras que en algunos casos los bits del flujo de bits recibidos en el demux 99 se pueden alternar de uno en uno entre cada uno de los descodificadores de vídeo (por ejemplo, los descodificadores de vídeo 30A y 30B en el ejemplo de la **FIG. 3B**), en muchos casos el flujo de bits se divide de manera diferente. Por ejemplo, el flujo de bits se puede dividir alternando el descodificador de vídeo que recibe el flujo de bits de bloque en bloque. En otro ejemplo, el flujo de bits se puede dividir por una relación de bloques no 1:1 para cada uno de los descodificadores de vídeo 30A y 30B. Por ejemplo, se pueden proporcionar dos bloques al descodificador de vídeo 30B para cada bloque proporcionado al descodificador de vídeo 30A. En algunos modos de realización, la división del flujo de bits por el demux 99 se puede preprogramar. En otros modos de realización, el demux 99 puede dividir el flujo de bits en base a una señal de control recibida desde un sistema externo al descodificador de vídeo 31, tal como desde un procesador del dispositivo de destino 14. La señal de control se puede generar en base a la resolución o la velocidad de bits de un vídeo desde la interfaz de entrada 28, en base a un ancho de banda del canal 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una resolución obtenible por el descodificador de vídeo 31.

Estructuras de codificación de vídeo escalable (SVC)

[0122] La **FIG. 4** es un diagrama conceptual que muestra unos ejemplos de escalabilidades en dimensiones diferentes. Como se analiza anteriormente, un ejemplo de implementación de SVC se refiere a la ampliación de codificación de vídeo escalable de HEVC. La ampliación de codificación de vídeo escalable de HEVC permite proporcionar información de vídeo en capas. Cada capa puede proporcionar información de vídeo correspondiente a una escalabilidad diferente. En HEVC, las escalabilidades se habilitan en tres dimensiones: escalabilidad

temporal (o de tiempo), escalabilidad espacial y escalabilidad de calidad (a veces denominada relación señal/ruido o escalabilidad de SNR). Por ejemplo, en la dimensión de tiempo, se pueden admitir frecuencias de tramas con 7,5 Hz, 15 Hz o 30 Hz, etc. mediante escalabilidad temporal (T). Cuando se admite escalabilidad espacial (S), se pueden permitir diferentes resoluciones, como QCIF, CIF y 4CIF, etc. Para cada resolución espacial y frecuencia de tramas específicas, se pueden añadir las capas de SNR (Q) para mejorar la calidad de la imagen.

[0123] Una vez que el contenido de vídeo se ha codificado de dicha manera escalable, se puede usar una herramienta extractora para adaptar el contenido real suministrado de acuerdo con unos requisitos de la aplicación, que pueden depender, por ejemplo, de los clientes o del canal de transmisión. En el ejemplo mostrado en la FIG. 4, cada cubo contiene las imágenes con la misma frecuencia de trama (nivel temporal), resolución espacial y capas de SNR. Por ejemplo, los cubos 402 y 404 contienen imágenes que tienen la misma resolución y SNR, pero diferentes frecuencias de tramas. Los cubos 402 y 406 representan imágenes que tienen la misma resolución (por ejemplo, en la misma capa espacial), pero diferentes SNR y frecuencias de tramas. Los cubos 402 y 408 representan imágenes que tienen la misma SNR (por ejemplo, en la misma capa de calidad), pero diferentes resoluciones y frecuencias de tramas. Los cubos 402 y 410 representan imágenes que tienen diferentes resoluciones, frecuencias de tramas y SNR. Se puede lograr una mejor representación añadiendo esos cubos (imágenes) en cualquier dimensión. Se admite la escalabilidad combinada cuando se permiten dos, tres o incluso más escalabilidades. Por ejemplo, combinando las imágenes del cubo 402 con las del cubo 404, se puede obtener una frecuencia de tramas más alta. Combinando las imágenes del cubo 404 con las del cubo 406, se puede lograr una mejor SNR.

[0124] De acuerdo con la ampliación SVC de HEVC, las imágenes con las capas espacial y de calidad más bajas son compatibles con H.264/AVC, y las imágenes en el nivel temporal más bajo forman la capa base temporal, que se puede mejorar con imágenes de niveles temporales más altos. Además de la capa compatible con HEVC, se pueden añadir varias capas de mejora espacial y/o de SNR para proporcionar escalabilidad espacial y/o de calidad. La escalabilidad de SNR también se denomina escalabilidad de calidad. Cada capa de mejora espacial o de SNR puede ser temporalmente escalable, con la misma estructura de escalabilidad temporal que la capa compatible con HEVC. Para una capa de mejora espacial o de SNR, la capa inferior de la que depende también se denomina capa base de esa capa de mejora espacial o de SNR específica.

[0125] La FIG. 5 es un diagrama conceptual que muestra un ejemplo de flujo de bits codificado de vídeo escalable. En el ejemplo de estructura de codificación de SVC mostrado en la FIG. 5, las imágenes con la capa espacial y de calidad más bajas (imágenes de la capa 502 y capa 504 que proporcionan resolución QCIF) son compatibles con HVC. Entre ellas, las imágenes del nivel temporal más bajo forman la capa base temporal 502, como se muestra en la FIG. 5. Esta capa base temporal (por ejemplo, la capa 502) se puede mejorar con imágenes de niveles temporales más altos, tal como la capa 504. Además de la capa compatible con HEVC, se pueden añadir varias capas de mejora espacial y/o de SNR para proporcionar escalabilidad espacial y/o de calidad. Por ejemplo, una capa de mejora puede ser una representación CIF que tiene la misma resolución que la capa 506. En el ejemplo mostrado en la FIG. 5, la capa 508 es una capa de mejora de SNR. Como se muestra en el ejemplo, cada capa de mejora espacial o de SNR puede ser temporalmente escalable, con la misma estructura de escalabilidad temporal que la capa compatible con HVC. Además, una capa de mejora puede mejorar tanto la resolución espacial como la frecuencia de tramas. Por ejemplo, la capa 510 proporciona una capa de mejora 4CIF, que incrementa aún más la frecuencia de tramas de 15 Hz a 30 Hz.

[0126] La FIG. 6 es un diagrama conceptual que muestra unos ejemplos de unidades de acceso (por ejemplo, una imagen codificada constituida por uno o más sectores) en un flujo de bits codificado de vídeo escalable 600. Como se muestra en la FIG. 6, en algunos modos de realización, los sectores codificados del mismo instante de tiempo son consecutivos en el orden de flujo de bits y forman una unidad de acceso en el contexto de SVC. Esas unidades de acceso SVC siguen entonces el orden de descodificación, que podría ser diferente del orden de visualización. El orden de descodificación se puede decidir, por ejemplo, mediante la relación de predicción temporal. Por ejemplo, la unidad de acceso 610 que consiste en la totalidad de las cuatro capas 612, 614, 616 y 618 para la trama 0 (por ejemplo, para la trama 0 como se ilustra en la FIG. 5) puede ir seguida de la unidad de acceso 620 que consiste en la totalidad de las cuatro capas 622, 624, 626 y 628 para la trama 4 (por ejemplo, para la trama 4 en la FIG. 5). La unidad de acceso 630 para la trama 2 puede seguir de forma desordenada, al menos desde una perspectiva de reproducción de vídeo. Sin embargo, la información de las tramas 0 y 4 se puede usar al codificar o descodificar la trama 2, y por lo tanto la trama 4 se puede codificar o descodificar antes de la trama 2. Las unidades de acceso 640 y 650 para las tramas restantes entre las tramas 0 y 4 pueden ser las siguientes, como se muestra en la FIG. 6.

Características de la codificación de vídeo escalable (SVC)

Predicción entre capas

[0127] Como se analiza anteriormente, en SVC, una capa de mejora se puede predecir en base a la información obtenida de una capa de referencia. Dicho procedimiento de predicción se denomina predicción entre capas. La predicción entre capas aprovecha las redundancias que existen entre las diferentes capas. Uno de los sistemas

de predicción entre capas es la intrapredicción entre capas. El modo de codificación que usa la intrapredicción entre capas se puede denominar modo "intra-BL". Dicho modo de predicción se ilustra a continuación con referencia a la **FIG. 7**.

5 **[0128]** La **FIG. 7** ilustra un esquema de un ejemplo 700 de predicción intra-BL. En particular, un bloque de capa base 712 de una capa base 710 está colocalizado con un bloque de capa de mejora 722 de una capa de mejora 720. En el modo intra-BL, la textura del bloque 722 se puede predecir usando la textura del bloque de capa base 712 colocalizado. Por ejemplo, es posible que unos valores de píxel del bloque de capa base 712 colocalizado y los valores de píxel del bloque de capa de mejora 722 sean muy similares entre sí, ya que el bloque de capa base 712 colocalizado esencialmente representa el mismo objeto de vídeo que el bloque de capa de mejora 722. Por tanto, los valores de píxel del bloque de capa base 712 colocalizado pueden servir como un predictor para predecir los valores de píxel del bloque de capa de mejora 722. El bloque de capa base 712 se puede sobremuestrear antes de usarse para predecir el bloque de capa de mejora 722 si la capa de mejora 720 y la capa base 710 tienen resoluciones diferentes. Por ejemplo, la imagen de la capa base puede ser de 1280×720 y la capa de mejora puede ser de 1920×1080, en cuyo caso el bloque de capa base o la imagen de capa base se pueden sobremuestrear por un factor de 1,5 en cada dirección (por ejemplo, horizontal y vertical) antes de usarse para predecir el bloque o la imagen de capa de mejora. El error de predicción (por ejemplo, residuo) se puede transformar, cuantificar y codificar entrópicamente. El término "colocalizado" se puede usar en el presente documento para describir la posición del bloque de capa base que representa el mismo objeto de vídeo que el bloque de capa de mejora. De forma alternativa, el término puede significar que el bloque de capa base colocalizado puede tener los mismos valores de coordenadas (después de tener en cuenta la relación de resolución entre la capa base y la capa de mejora) que el bloque de capa de mejora. Aunque en esta divulgación se usa el término "colocalizado", se pueden aplicar técnicas similares a bloques vecinos (por ejemplo, adyacentes) del bloque actual, a bloques vecinos (por ejemplo, adyacentes) del bloque colocalizado del bloque actual, o a otros bloques relacionados cualesquiera.

10
15
20
25 **[0129]** Otro enfoque para la predicción de textura entre capas puede implicar el uso de una imagen de referencia entre capas (ILRP). En dicho ejemplo, se inserta una imagen de capa base reconstruida (tras el sobremuestreo necesario) en la lista de imágenes de referencia de la imagen de capa de mejora correspondiente. La predicción de textura entre capas se logra cuando la capa de mejora se predice usando la imagen de referencia entre capas.

30 **Escalabilidad**

[0130] Los sistemas de codificación de vídeo escalable pueden proporcionar diversas escalabilidades, tal como la escalabilidad espacial y la escalabilidad de profundidad de bits. Como se analiza anteriormente, la codificación de vídeo escalable proporciona una o más capas de mejora escalables que se pueden descodificar en combinación con la capa base para lograr mayores niveles espaciales, temporales y/o de señal-ruido (SNR).

35
40 **[0131]** La escalabilidad espacial se refiere al caso en el que las imágenes de capa base y las imágenes de capa de mejora tienen diferentes tamaños. Por ejemplo, las imágenes de capa base pueden tener el tamaño de 1280 píxeles por 720 píxeles, mientras que las imágenes de la capa de mejora pueden tener el tamaño de 1920 píxeles por 1080 píxeles.

45 **[0132]** La escalabilidad de profundidad de bits se refiere al caso en el que las imágenes de capa base y las imágenes de capa de mejora tienen diferentes profundidades de bits. Por ejemplo, las imágenes de capa base pueden tener una profundidad de bits de 8 bits (por ejemplo, los componentes de color se representan con 8 bits, dando como resultado $2^8 = 256$ valores totales posibles), mientras que las imágenes de capa de mejora pueden tener una profundidad de bits de 10 bits (por ejemplo, los componentes de color se representan con 10 bits, dando como resultado $2^{10} = 1024$ valores totales posibles). También es posible que un componente de color (por ejemplo, luma) se represente usando una profundidad de bits, y otro componente de color (por ejemplo, croma) se represente usando otra profundidad de bits.

50
55 **[0133]** Usando SVC para generar un flujo de bits escalable que contiene una capa base que un descodificador heredado (por ejemplo, de 720 p y/u 8 bits) puede descodificar para generar un contenido de vídeo que tiene una profundidad de bits menor (por ejemplo, de 720 p y/u 8 bits), y una o más capas de mejora que un descodificador escalable puede descodificar para generar un contenido de vídeo más mejorado (por ejemplo, de 1080 p y/o 10 bits), se puede proporcionar retrocompatibilidad con los descodificadores heredados y se pueden reducir los requisitos de ancho de banda en comparación con los flujos de bits de transmisión simultánea separados, mejorando de este modo la eficacia y el rendimiento de la codificación.

60 **Escalabilidad espacial y predicción entre capas**

65 **[0134]** Como se analiza anteriormente, los valores de píxel de la capa base se pueden usar para predecir los valores de píxel de la capa de mejora. En el caso de la escalabilidad espacial, las imágenes de la capa base y las imágenes de la capa de mejora tienen diferentes tamaños, por lo que tal vez sea necesario modificar las imágenes de capa base (por ejemplo, de modo que tengan la misma resolución que las imágenes de capa de mejora) antes de usarlas para predecir las imágenes de capa de mejora. Por ejemplo, las imágenes de capa base se pueden

sobremuestrear (por ejemplo, si el tamaño de la capa de mejora es mayor que el de la capa base) mediante la relación de resolución entre la capa base y la capa de mejora.

[0135] A continuación, se ilustra un ejemplo de dicha modificación en el caso de la escalabilidad espacial. En este ejemplo, una implementación que usa un filtro de sobremuestreo de 2 tomas se usa para estimar el valor del componente de color P_{ei} de un píxel de capa de mejora en base a los valores de componente de color P_{bi0} y P_{bi1} de los píxeles de capa base. Aquí, la predicción entre capas se puede lograr usando pesos asociados a los valores de píxel de capa base (o valores de componentes de color de los mismos). Una de dichas relaciones se expresa en la siguiente ecuación, donde P'_{ei} representa el valor predicho de P_{ei} :

$$P'_{ei} = (w_0 P_{bi0} + w_1 P_{bi1} + O) \gg T \quad (1)$$

[0136] En este ejemplo, w_0 y w_1 representan pesos y O representa el desplazamiento del filtro de sobremuestreo. Por ejemplo, los pesos pueden ser factores de peso. En este ejemplo, los valores de componente de color (por ejemplo, componente de luma) P_{bi0} y P_{bi1} se multiplican por los pesos w_0 y w_1 , respectivamente. En un modo de realización, la suma de todos los pesos (por ejemplo, pesos w_i) del filtro de sobremuestreo es igual a $(1 \ll T)$, que denota un 1 binario con un desplazamiento a la izquierda de T bits. El valor de T se puede elegir o determinar en base a la cantidad de exactitud deseada. Si en un modo de realización se desea más exactitud, el valor de T se puede incrementar. Un valor incrementado de T significa que se realiza más desplazamiento de bits a la izquierda, dando como resultado más bits para realizar cálculos. En un modo de realización, el valor del desplazamiento O es $(1 \ll (T-1))$. En otro modo de realización, el valor del desplazamiento O puede ser la mitad del valor de la suma de todos los pesos (por ejemplo, w_i). Por ejemplo, si el valor de T es igual a 4, la suma de todos los coeficientes sería $1 \ll 4$, que es $(10000)_2 = 16$. En el mismo ejemplo, el desplazamiento O sería $1 \ll 3$, que es $(1000)_2 = 8$. El desplazamiento O puede ser un desplazamiento de redondeo de modo que los cálculos de la ecuación (1) se redondean al alza en lugar de redondearse a la baja. En algunos modos de realización, el desplazamiento puede ser cero.

Escalabilidad de profundidad de bits y predicción entre capas

[0137] De forma similar, en el caso de la escalabilidad de profundidad de bits, los píxeles de capa base y los píxeles de capa de mejora se representan en diferentes números de bits (por ejemplo, 8 bits frente a 10 bits), por lo que tal vez sea necesario modificar los valores de píxel de capa base (por ejemplo, someterlos a una conversión de profundidad de bits para que tengan el mismo número de bits que los valores de píxel la capa de mejora) antes de utilizarlos para predecir los valores de píxel de capa de mejora. Un ejemplo de dicha modificación (por ejemplo, la conversión de profundidad de bits) implica aplicar un desplazamiento a la izquierda de N a los bits de los píxeles de capa base (por ejemplo, si la capa de mejora tiene una mayor profundidad de bits), donde N representa la diferencia de profundidad de bits de la capa base y la capa de mejora.

[0138] A continuación, se ilustra un ejemplo de dicha modificación en el caso de la escalabilidad de profundidad de bits. En este ejemplo, M representa la profundidad de bits de la capa de mejora, N representa la profundidad de bits de la capa base y K representa la diferencia de profundidad de bits, que es $M - N$. Aquí, la profundidad de bits se refiere a profundidad de bits de un componente de color en particular, como Y , U o V en el ejemplo del espacio de color YUV. En este ejemplo, el componente de luma se usa para ilustrar el modo de realización, pero se pueden aplicar procedimientos similares a otros componentes de color.

[0139] En este ejemplo, P_{ei} representa el componente de luma de un píxel de la capa de mejora, y P_{bi} representa el componente de luma de un píxel correspondiente (por ejemplo, colocalizado) de la capa base. Aquí, la predicción de textura entre capas puede implicar convertir los componentes de color de capa base de modo que estén representados en el mismo número de bits que los componentes de color de capa de mejora. Una implementación de dicha conversión se ilustra en las siguientes ecuaciones, donde P'_{ei} representa el valor predicho de P_{ei} :

$$P'_{ei} = P_{bi} \ll K, \text{ if } K \geq 0 \quad (2)$$

$$P'_{ei} = P_{bi} \gg (-K), \text{ if } K < 0 \quad (3)$$

[0140] Por ejemplo, si la diferencia de profundidad de bits entre la capa base y la capa de mejora es 2 (por ejemplo, la capa base tiene una profundidad de bits de 8 bits y la capa de mejora tiene una profundidad de bits de 10 bits), el componente de luma del píxel de capa base se desplaza 2 bits a la izquierda. Por tanto, si un píxel (por ejemplo, de color blanco) de la capa base tiene un componente de luma de 255 (1111 1111 en binario) en la capa base, de acuerdo con la ecuación (1), el valor predicho sería 1020 (11 1111 1100 en binario), que está en la vecindad del componente de luma del color blanco en una profundidad de bits de 10 bits, 1023.

[0141] En la presente divulgación, se supone que K es mayor o igual que 0 (por ejemplo, la profundidad de bits de la capa de mejora es mayor o igual que la profundidad de bits de la capa base). Sin embargo, se pueden aplicar procedimientos similares al caso en el que K es menor que 0.

Escalabilidad espacial y de profundidad de bits combinada

5 [0142] Si en un modo de realización están presentes tanto la escalabilidad espacial como la escalabilidad de profundidad de bits, los procedimientos analizados anteriormente se pueden combinar para lograr tanto el sobremuestreo como la conversión de profundidad de bits para predecir los valores de píxel de capa de mejora. Por ejemplo, una implementación puede ser:

- 10 1. Desplazar K bits a la izquierda el píxel de capa base P_{bli} : $P'_{bli} = P_{bli} \ll K$
 2. Sobremuestrear el píxel desplazado a la izquierda P'_{bli} : $P'_{el} = (w_0P'_{bli0} + w_1P'_{bli1} + O) \gg T$

15 [0143] En el ejemplo mostrado anteriormente, el proceso de conversión de profundidad de bits (por ejemplo, el primer proceso) realiza la conversión de profundidad de bits en los valores de píxel de capa base y facilita valores de píxel con conversión en profundidad de bits, y el proceso de sobremuestreo (por ejemplo, el segundo proceso) realiza un sobremuestreo en los valores de píxel con conversión de profundidad de bits y facilita valores de píxel sobremuestreados con conversión de profundidad de bits. Por tanto, cuando los procesos se realizan en serie, se puede lograr tanto el sobremuestreo como la conversión de profundidad de bits.

20 [0144] En otro modo de realización, el píxel de capa base se puede sobremuestrear primero de acuerdo con la relación de resolución, y a continuación desplazarse K bits a la izquierda para obtener el valor de píxel de capa de mejora predicho P'_{el} (por ejemplo, un valor de píxel sobremuestreado con conversión de profundidad de bits).

Proceso de una sola fase para escalabilidad espacial y de profundidad de bits

25 [0145] En el ejemplo analizado anteriormente, el sobremuestreo y la conversión de profundidad de bits se llevan a cabo como un proceso de dos fases. En otro modo de realización, el sobremuestreo y la conversión de profundidad de bits se realizan en un proceso de una sola fase para generar una predicción para la escalabilidad de profundidad de bits y espacial combinada. En la siguiente ecuación se ilustra una implementación de dicho proceso de una sola fase.

$$P'_{el} = (w_0P_{bli0} + w_1P_{bli1} + O) \gg W, \text{ donde } W = T - K \quad (4)$$

35 [0146] La ecuación (4) ilustra un proceso de sobremuestreo, en el que el desplazamiento a la derecha se reduce en la diferencia de profundidad de bits K entre la capa base y la capa de mejora. Por tanto, realizando un desplazamiento de bits a la derecha de K bits menos que en el caso sin escalabilidad de profundidad de bits, el proceso ilustrado en la ecuación (4) realiza eficazmente el desplazamiento de bits a la izquierda de K bits, mientras lleva a cabo el sobremuestreo al mismo tiempo. En el ejemplo de la ecuación (4), el proceso de una sola fase ilustrado en la ecuación (4) se aplica a los valores de píxel de capa base y facilita valores de píxel sobremuestreados con conversión de profundidad de bits. Esto es diferente del ejemplo descrito anteriormente, donde el proceso de conversión de profundidad de bits se realiza en los valores de píxel de capa base, y el proceso de sobremuestreo se realiza en la salida del proceso de conversión de profundidad de bits (por ejemplo, los valores de píxel de capa base con conversión de profundidad de bits).

45 [0147] En algunos modos de realización, el sobremuestreo y la conversión de profundidad de bits se realizan simultáneamente. En un modo de realización, realizar el sobremuestreo y la conversión de profundidad de bits "simultáneamente" puede significar que hay al menos una operación que contribuye tanto al sobremuestreo como a la conversión de profundidad de bits. En otro modo de realización, realizar simultáneamente el sobremuestreo y la conversión de profundidad de bits puede significar que se usa un solo filtro para realizar tanto el sobremuestreo como la conversión de profundidad de bits. En otro modo de realización, realizar simultáneamente el sobremuestreo y la conversión de profundidad de bits puede significar que el sobremuestreo y la conversión de profundidad de bits se superponen temporalmente.

55 [0148] En algunos modos de realización, en lugar de terminar el proceso de sobremuestreo antes de realizar un proceso de conversión de profundidad de bits por separado, el proceso de conversión de profundidad de bits se puede integrar en un proceso de sobremuestreo de una sola fase. De forma alternativa, en algunos modos de realización, en lugar de terminar el proceso de conversión de profundidad de bits antes de realizar un proceso de sobremuestreo separado, el proceso de sobremuestreo se puede integrar en un proceso de conversión de profundidad de bits de una sola fase. En un modo de realización, el proceso de una sola fase puede incluir múltiples operaciones matemáticas tales como una multiplicación, una suma y/o un desplazamiento de bits. El proceso de una sola fase puede incluir al menos una operación que contribuye tanto al sobremuestreo como a la conversión de profundidad de bits. En un ejemplo, dicha operación puede ser un desplazamiento de bits a la izquierda. En otro ejemplo, dicha operación puede ser un desplazamiento de bits a la derecha. En otro ejemplo más, dicha operación puede implicar una multiplicación. En otro ejemplo más, dicha operación puede implicar una suma.

65

[0149] En un modo de realización, después de realizar el proceso de sobremuestreo, tal vez no sea necesario realizar un proceso de conversión de profundidad de bits por separado, porque la conversión de profundidad de bits ya se ha realizado como parte del proceso de sobremuestreo. Por ejemplo, el proceso de conversión de profundidad de bits se puede integrar en el proceso de sobremuestreo. En un modo de realización, una de las operaciones realizadas en relación con el proceso de sobremuestreo también logra la conversión de profundidad de bits, eliminando la necesidad de realizar un proceso de conversión de profundidad de bits por separado.

[0150] En algunos modos de realización, al realizar el sobremuestreo y la conversión de profundidad de bits en un proceso de una sola fase, se puede reducir el error de redondeo en el proceso de sobremuestreo. Por ejemplo, en el proceso de una sola fase mostrado anteriormente, se aplica un desplazamiento de W a la derecha a la suma de los valores de píxel ponderados w_0P_{b10} y w_1P_{b11} y el desplazamiento O , que es menor que T en la ecuación (1), de tal modo que se conservan más dígitos, dando lugar de este modo a una mejor exactitud.

[0151] La FIG. 8 es un diagrama de flujo que ilustra un procedimiento 800 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la FIG. 8 se pueden realizar mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la FIG. 2A o la FIG. 2B), un decodificador (por ejemplo, el decodificador de vídeo como se muestra en la FIG. 3A o la FIG. 3B), o cualquier otro componente. Por conveniencia, el procedimiento 800 que se describe se realiza mediante un codificador, que puede ser el codificador, el decodificador u otro componente.

[0152] El procedimiento 800 comienza en el bloque 801. En el bloque 805, el codificador almacena información de vídeo asociada a una capa de mejora (EL) que comprende una unidad de vídeo EL y una capa de referencia (RL) que comprende una unidad de vídeo RL correspondiente a la unidad de vídeo EL. En el bloque 810, el codificador realiza el sobremuestreo y la conversión de profundidad de bits en la información de píxeles de la unidad de vídeo RL para determinar la información de píxeles predichos de la unidad de vídeo EL. Por ejemplo, el codificador realiza simultáneamente el sobremuestreo y la conversión de profundidad de bits aplicando un proceso de una sola fase. Dicho proceso de una sola fase puede incluir al menos una operación que contribuye tanto al sobremuestreo como a la conversión de profundidad de bits. En un modo de realización, la información de píxeles se refiere a unos valores de píxel o componentes de color de dichos valores de píxel, y la información de píxeles predichos se refiere al predictor para determinar los valores de píxel o componentes de color de la unidad de vídeo EL. En un modo de realización, la información de predicción de píxeles se puede determinar aplicando un filtro combinado de sobremuestreo y conversión de profundidad de bits configurado para realizar el sobremuestreo y el desplazamiento de bits de valores de píxel en un proceso de una sola fase (por ejemplo, simultáneamente). En el bloque 815, el codificador determina la unidad de vídeo EL usando la información de píxeles predichos. Por ejemplo, dicho proceso puede implicar restar el (los) valor(es) de predicción obtenido(s) aplicando la función de predicción al (a los) valor(es) de píxel RL del (de los) valor(es) reales del bloque EL, y transmitir el residuo y la predicción. El procedimiento 800 termina en el bloque 820.

[0153] Como se analiza anteriormente, uno o más componentes de los codificadores de vídeo 20 de la FIG. 2A, el codificador de vídeo 21 de la FIG. 2B, el decodificador de vídeo 30 de la FIG. 3A, o el decodificador de vídeo 31 de la FIG. 3B (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se pueden usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, tales como realizar un sobremuestreo y una conversión de bits de profundidad en la información de píxeles, y determinar la unidad de vídeo EL (por ejemplo, el bloque actual de la EL) usando la información de píxeles predichos.

[0154] La FIG. 9 es un diagrama de flujo que ilustra un procedimiento 900 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la FIG. 9 se pueden realizar mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la FIG. 2A o la FIG. 2B), un decodificador (por ejemplo, el decodificador de vídeo como se muestra en la FIG. 3A o la FIG. 3B) o cualquier otro componente. Por conveniencia, el procedimiento 900 que se describe se realiza mediante un codificador, que puede ser el codificador, el decodificador u otro componente.

[0155] El procedimiento 900 comienza en el bloque 901. En el bloque 905, el codificador determina si las resoluciones de la capa de referencia y la capa de mejora son diferentes. Si el codificador determina que las resoluciones son diferentes, el codificador determina, en el bloque 910, si las profundidades de bits de la capa de referencia y la capa de mejora son diferentes. Si el codificador determina que las profundidades de bits son diferentes, el codificador realiza simultáneamente un sobremuestreo y una conversión de profundidad de bits en el bloque 925 (por ejemplo, en la información de píxeles del bloque RL o la imagen RL, como se analiza con referencia a la FIG. 8), para determinar la información de píxeles predichos del bloque actual en la capa de mejora. Si el codificador determina en el bloque 910 que las profundidades de bits no son diferentes, el codificador realiza un sobremuestreo en el bloque 920 para determinar información de píxeles predichos del bloque actual. Si el codificador determina en el bloque 905 que las resoluciones no son diferentes, el codificador determina, en el bloque 915, si las profundidades de bits son diferentes. Si el codificador determina que las profundidades de bits son diferentes, el codificador realiza una conversión de profundidad de bits en el bloque 930 para determinar la información de píxeles predichos del bloque actual. Si el codificador determina en el bloque 915 que las profundidades de bits no son diferentes, el codificador determina el bloque actual en base a la información de

píxeles predichos (por ejemplo, la información de píxeles del bloque colocalizado en la capa de referencia) en el bloque 935. De forma similar, después de que el codificador determine la información de píxeles predichos en los bloques 920, 925 o 930, el codificador determina el bloque actual en base a la información de píxeles procesada del bloque colocalizado en la capa de referencia (por ejemplo, información de píxeles predichos) en el bloque 935. El procedimiento 900 termina en el bloque 940.

[0156] Como se analiza anteriormente, uno o más componentes del codificador de vídeo 20 de la FIG. 2A, el codificador de vídeo 21 de la FIG. 2B, el decodificador de vídeo 30 de la FIG. 3A, o el decodificador de vídeo 31 de la FIG. 3B (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se pueden usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, tales como determinar si las resoluciones y/o las profundidades de bits son diferentes, realizar un sobremuestreo, una conversión de profundidad de bits o un sobremuestreo y una conversión de profundidad de bits simultáneos, y determinar el bloque actual de la EL usando la información de píxeles predichos.

[0157] En el procedimiento 900, uno o más de los bloques mostrados en la FIG. 9 se pueden eliminar (por ejemplo, no realizar) y/o el orden en el que se realiza el procedimiento se puede cambiar. Por ejemplo, aunque se comprueba si las resoluciones son diferentes antes de comprobar si las profundidades de bits son diferentes en el ejemplo de la FIG. 9, en otro ejemplo, se puede comprobar si la profundidad de bits es diferente antes de comprobar si las resoluciones son diferentes.

Recorte del valor predicho

[0158] En un modo de realización, se puede aplicar un proceso de recorte a la predicción mostrada en la ecuación (4) para limitar el intervalo de bits de los píxeles de predicción como se muestra:

$$P'_{el} = \text{CLIP}(w_0P_{b10} + w_1P_{b11} + O) \gg W, \text{ donde } W = T - K \quad (5)$$

[0159] En este ejemplo, el valor de P'_{el} se puede recortar hasta un valor del intervalo $[0, (1 \ll M) - 1]$ (por ejemplo, para evitar el subdesbordamiento o el desbordamiento). Por ejemplo, si la profundidad de bits de la EL es 10, el valor predicho P'_{el} se puede recortar hasta el intervalo $[0, 1023]$. El valor mínimo y el valor máximo no están limitados a dicho ejemplo, pudiendo el codificador predefinirlos o señalarlos. La profundidad de bits (o el intervalo de valores) se puede indicar en el PPS.

Filtros bidimensionales para escalabilidad espacial y de profundidad de bits

[0160] El ejemplo analizado con referencia a la ecuación (4) ilustra el caso de un filtro de sobremuestreo unidimensional (1D). En el caso en el que las imágenes de la capa base y la capa de mejora comprenden matrices de píxeles 2D, dicho filtro 1D se puede usar para sobremuestrear los píxeles en la dirección vertical (por ejemplo, P_{b10} y P_{b11} son píxeles vecinos en sentido vertical en la capa base) o en la dirección horizontal (por ejemplo, P_{b10} y P_{b11} son píxeles vecinos en sentido horizontal en la capa base). Usando el filtro de sobremuestreo 1D, se pueden generar valores de predicción para píxeles en todas las direcciones.

[0161] En un modo de realización, se puede usar un filtro separable bidimensional. Por ejemplo, se puede aplicar un filtro de sobremuestreo 1D horizontal, y a continuación, se puede aplicar un filtro de sobremuestreo 1D vertical. En otro modo de realización, el filtro de sobremuestreo 1D vertical se puede aplicar antes de aplicar el filtro de sobremuestreo 1D horizontal. En el ejemplo en el que se selecciona el sobremuestreo horizontal como la primera fase y se selecciona el sobremuestreo vertical como la segunda fase, la cantidad de desplazamiento a la derecha en la segunda fase (por ejemplo, el sobremuestreo vertical) se puede reducir para reflejar la diferencia de profundidad de bits entre la capa base y la capa de mejora. Por ejemplo, si T_{sec} representa la cantidad de desplazamiento a la derecha en la segunda fase en ausencia de cualquier escalabilidad de profundidad de bits, en el filtro combinado de sobremuestreo y conversión de profundidad de bits (por ejemplo, el filtro 2D separable analizado anteriormente), la cantidad de desplazamiento a la derecha de $T_{sec} - K$ se puede usar en la segunda fase. En otras palabras, la cantidad de desplazamiento a la derecha se reduce en la diferencia de profundidad de bits entre la capa base y la capa de mejora, de modo que se elimina la necesidad de desplazar a la izquierda el valor predicho en una fase separada. Por ejemplo, si T_{sec} se establece en 4 y la diferencia de profundidad de bits es 2, en la segunda fase, el desplazamiento a la derecha de la suma de los componentes de color ponderados es $4 - 2 = 2$, en lugar de 4. Aunque el sobremuestreo vertical se selecciona como la segunda fase en este ejemplo, se puede aplicar la técnica similar cuando se selecciona el sobremuestreo horizontal como la segunda fase. En un modo de realización, el desplazamiento de bits en la última fase del filtro 2D (o de otra dimensión) separable se reduce en la diferencia de profundidad de bits de la capa base y la capa de mejora.

[0162] En otro modo de realización, el sobremuestreo se puede realizar mediante un filtro 2D no separable, como se muestra a continuación:

$$P'_{el} = \text{Sum}(w_{ij} * P_{bij}) + O \gg T \quad (6)$$

donde P_{bij} es el píxel en la ubicación (i, j) y w_{ij} es el peso correspondiente para P_{bij} . En un modo de realización, la suma de todos los pesos w_{ij} es igual a $1 \ll T$. La combinación de escalabilidad espacial (con filtro de sobremuestreo 2D no separable) y escalabilidad de profundidad de bits se puede implementar usando la ecuación siguiente:

$$P'_{cl} = \text{Sum} (w_{ij} * P_{bij}) + O \gg W, \text{ donde } W = T - K \quad (7)$$

[0163] Como se analiza anteriormente, al realizar el sobremuestreo y la conversión de profundidad de bits en un proceso de una sola fase, el error de redondeo se puede reducir, logrando de ese modo una mejor exactitud.

Señalización adaptativa de coeficientes de filtro

[0164] En un modo de realización del sistema combinado de profundidad de bits y escalabilidad espacial, se puede usar el sobremuestreo/filtrado y/o el filtrado entre componentes y entre capas y/o el filtrado entre componentes adaptativo. Los coeficientes de filtro se pueden señalar adaptativamente en el flujo de bits, tal como en un PPS, un APS, una cabecera de sector y las ampliaciones relacionadas. Por ejemplo, para algunas tramas (o sectores), los coeficientes del filtro se pueden señalar (por ejemplo, transmitir al descodificador), y para algunas otras tramas, los coeficientes de filtro pueden adoptar uno o más valores predefinidos. Dichos valores predefinidos pueden ser conocidos por el codificador y/o el descodificador. Por tanto, la señalización o la determinación de los coeficientes de filtro puede diferir para diferentes tramas (o sectores). Por ejemplo, se puede determinar si se señalizan y cómo se señalizan los coeficientes de filtro en base a una información secundaria, que puede incluir uno o más, aunque no de forma limitativa, de entre un espacio de color, formato de color (4:2:2, 4:2:0, etc.), tamaño de trama, tipo de trama, modo de predicción, dirección de interpredicción, modo de intrapredicción, tamaño de unidad de codificación (CU), tamaño de unidad de codificación máximo/mínimo, parámetro de cuantificación (QP), tamaño de unidad de transformada máximo/mínimo (TU), índice de trama de referencia de profundidad de árbol de transformada máximo, ID de capa temporal, etc. Por ejemplo, los coeficientes de filtro se pueden transmitir solo para tramas que tienen un tamaño mayor que un tamaño umbral. En otro modo de realización, los coeficientes de filtro se pueden transmitir solo para tramas que tienen un tamaño menor que un tamaño umbral.

[0165] En un modo de realización, puede haber dos o más conjuntos de coeficientes de filtro disponibles, y el codificador puede señalar un bit (o unos bits) elegido(s) que indica(n) qué conjunto de coeficientes de filtro se va a usar para la trama. Por tanto, diferentes tramas (o sectores) pueden usar diferentes conjuntos de coeficientes de filtro. Por ejemplo, una trama puede incluir bordes de alto contraste, y se pueden usar coeficientes de filtro que generan un efecto de paso bajo mínimo. En el mismo ejemplo, otra trama puede incluir mucho ruido, y se pueden usar coeficientes de filtro que generan un efecto de paso bajo más fuerte. En un modo de realización, el codificador predice la trama de la capa de mejora con cada conjunto disponible de coeficientes de filtro y selecciona un conjunto en base al rendimiento de codificación. En un modo de realización, se puede almacenar, en algún lugar (por ejemplo, una memoria), una pluralidad de conjuntos de coeficientes de filtro que el codificador y/o el descodificador pueden usar. En otro modo de realización, el codificador y/o el descodificador pueden obtener sobre la marcha los coeficientes de filtro. En otro modo de realización más, el codificador transmite los coeficientes de filtro al descodificador.

[0166] En un modo de realización, el filtro entre capas adaptativo y/o el filtro entre componentes y entre capas y/o el filtro entre componentes analizados anteriormente están habilitados y/o inhabilitados en el flujo de bits tal como en VPS, SPS y las ampliaciones relacionadas. Por ejemplo, los filtros adaptativos se pueden habilitar solo para un subconjunto de las tramas e inhabilitar para otras tramas.

Desplazamiento para proceso de una sola fase

[0167] En el ejemplo analizado con referencia a la ecuación (4), no se añade ningún desplazamiento después de realizar el desplazamiento de bits. En otro modo de realización, se añade un desplazamiento S después de que se haya realizado el desplazamiento de bits, como se ilustra a continuación:

$$P'_{cl} = ((w_0 P_{bl0} + w_1 P_{bl1} + O) \gg W) + S \quad (8)$$

[0168] En un modo de realización, el desplazamiento S puede ser $1 \ll (K - 1)$. En este ejemplo, el desplazamiento O se proporciona para el proceso de sobremuestreo y el desplazamiento S se proporciona para el proceso de conversión de profundidad de bits (por ejemplo, para acercar el valor predicho al valor de píxel de la capa de mejora real).

[0169] En otro ejemplo, se puede añadir un desplazamiento antes de realizar el desplazamiento de bits. En otras palabras, el desplazamiento S de la ecuación (8) se puede integrar en el desplazamiento O de la ecuación (8), como se ilustra a continuación:

$$P'_{cl} = (w_0 P_{bl0} + w_1 P_{bl1} + O') \gg W \quad (9)$$

donde el desplazamiento combinado O' representa el desplazamiento S integrado en el desplazamiento O de la ecuación (8). Por ejemplo, si el desplazamiento O tiene un valor de $1 \ll (T - 1)$ y el desplazamiento S tiene un valor de $1 \ll (K - 1)$ en el ejemplo de la ecuación (8), el nuevo desplazamiento combinado O' sería igual a $O + (S \ll W)$, que es $1 \ll T$. En este ejemplo, dado que el desplazamiento se añade en una fase intermedia (por ejemplo, antes de que realizar el desplazamiento de bits), se mantiene una mayor exactitud.

[0170] En un modo de realización, el recorte analizado con referencia a la ecuación (5) se aplica a la ecuación (8) o la ecuación (9). Por ejemplo, el valor de P'_{el} se puede recortar hasta un valor del intervalo $[0, (1 \ll M) - 1]$, y se puede evitar cualquier subdesbordamiento o desbordamiento.

Selección adaptativa de desplazamiento combinado

[0171] En un modo de realización, el desplazamiento combinado O' analizado anteriormente se puede seleccionar adaptativamente (por ejemplo, para mejorar el rendimiento de velocidad-distorsión). Dicha selección adaptativa del desplazamiento combinado O' se puede hacer, por ejemplo, en base a una unidad de árbol de codificación (CTU) o de imagen en imagen.

[0172] Por ejemplo, en una imagen de capa base de 8 bits, puede haber dos píxeles que representan dos colores diferentes: blanco y negro. En la imagen de capa base, el píxel correspondiente al color negro puede tener un valor de componente de color de cero, y el píxel correspondiente al color blanco puede tener un valor de componente de color de 255. Si estos dos píxeles de capa base se van a usar para predecir píxeles de capa de mejora de 10 bits, primero los píxeles se pueden desplazar 2 bits a la izquierda, para pasar de la representación de 8 bits a la representación de 10 bits. Cuando se realiza la conversión de profundidad de bits, el valor predicho del píxel negro sigue siendo 0 ($0 \ll 2$), y el valor predicho del píxel blanco es 1020 ($1111\ 1111 \ll 2$, que es $11\ 1111\ 1100 = 1020$). Después de que se haya realizado la conversión de profundidad de bits, el valor predicho del píxel negro es igual al valor real 0, pero el valor predicho del píxel blanco de 1020 tiene un desplazamiento de 3, ya que el valor de píxel de capa de mejora real sería de 1023. En un modo de realización, se añade un desplazamiento fijo a todos los valores predichos de modo que se puede reducir el error de predicción global. En un ejemplo, el desplazamiento puede ser el valor del centro del intervalo. Por ejemplo, se puede añadir un desplazamiento fijo de 2 a todos los valores predichos, con lo cual se reduciría el error de predicción promedio. En otro modo de realización, el valor de desplazamiento se selecciona adaptativamente de entre una pluralidad de valores de desplazamiento (por ejemplo, 0, 1, 2, 3 en este ejemplo) de modo que el error de predicción se reduce al mínimo para cada píxel (o cada PU, CU, LCU o imagen, o cualquier otra denominación). Por ejemplo, se puede seleccionar que el desplazamiento para el píxel negro sea 0, y se puede seleccionar que el desplazamiento para el píxel blanco sea 3.

Obtención de desplazamiento combinado a partir de una tabla de consulta

[0173] En un modo de realización, el desplazamiento combinado O' se puede obtener a partir de una tabla de consulta (por ejemplo, una tabla de consulta de correlación de intervalos). Dicha tabla de consulta puede tomar valores de píxel de capa base como entrada, y facilitar valores de desplazamiento para los valores de píxel de capa base respectivos. Por tanto, un valor de desplazamiento que se va a usar para predecir un píxel de capa de mejora puede estar basado en el valor del componente de color (por ejemplo, el valor de intensidad de píxel) del píxel correspondiente (por ejemplo, colocalizado) de la capa base. En un modo de realización, el codificador transmite una tabla de consulta al descodificador. En otro modo de realización, tanto el codificador como el descodificador conocen una tabla de consulta fija. En dicho caso, el codificador puede simplemente señalar al descodificador que el codificador está usando la tabla de consulta.

Predicción de componentes cruzados

[0174] En algunos modos de realización, se pueden procesar por separado diferentes componentes de color (por ejemplo, de luma, de croma, etc.). Por ejemplo, los componentes de luma de los píxeles de capa base se pueden usar para predecir componentes de croma de píxeles de capa de mejora. En otros modos de realización, se pueden usar uno o más componentes de color para predecir otros componentes de color. Por ejemplo, el componente de luma de un píxel de capa base se puede usar para predecir los K bits (diferencia de profundidad de bits) más bajos de los componentes de croma del píxel de capa de mejora correspondiente (y viceversa), como se ilustra a continuación:

$$P'_{el,Cb} = [(w_0 P_{bl0,Cb} + w_1 P_{bl1,Cb} + O') \gg W] + [(w_0' P_{bl0,Y} + w_1' P_{bl1,Y} + O'') \gg (T + N - K)] \quad (10)$$

donde W es igual a (T - K). Por ejemplo, la señal de luma puede incluir más detalles que las señales de croma. Por tanto, si los componentes de croma de la capa base se filtran para predecir componentes de croma de capa de mejora, se puede perder fácilmente una parte del detalle durante el proceso. En el ejemplo de la ecuación (10),

los componentes de luma se usan para predecir los componentes de croma. Por tanto, al hacer eso se puede mantener más detalle.

5 [0175] La información y las señales divulgadas en el presente documento se pueden representar usando cualquiera de entre una variedad de tecnologías y técnicas diferentes. Por ejemplo, los datos, instrucciones, mandatos, información, señales, bits, símbolos y chips que se pueden haber mencionado a lo largo de la descripción anterior se pueden representar mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticas, campos o partículas ópticas o cualquier combinación de los mismos.

10 [0176] Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en relación con los modos de realización divulgados en el presente documento se pueden implementar como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, se han descrito anteriormente diversos componentes, bloques, módulos, circuitos y etapas ilustrativas en general en términos de su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de la aplicación en particular y las restricciones de diseño impuestas al sistema global. Los expertos en la técnica pueden implementar la funcionalidad descrita de formas diferentes para cada aplicación en particular, pero no se debería interpretar que dichas decisiones de implementación suponen apartarse del alcance de la presente invención.

20 [0177] Las técnicas descritas en el presente documento se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Dichas técnicas se pueden implementar en cualquiera de entre una variedad de dispositivos tales como ordenadores de propósito general, terminales de dispositivos de comunicación inalámbrica o dispositivos de circuitos integrados que tienen múltiples usos, incluyendo su aplicación en terminales de dispositivos de comunicación inalámbrica y otros dispositivos. Características cualesquiera descritas como módulos o componentes se pueden implementar juntas en un dispositivo lógico integrado, o por separado como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas se pueden realizar, al menos en parte, mediante un medio de almacenamiento de datos legible por ordenador que comprende código de programa que incluye instrucciones que, cuando se ejecutan, realizan uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, tales como memoria de acceso aleatorio (RAM), tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de solo lectura programable y borrable eléctricamente (EEPROM), memoria FLASH, medios de almacenamiento de datos magnéticos u ópticos, y similares. Las técnicas se pueden realizar de forma adicional o de forma alternativa, al menos en parte, mediante un medio de comunicación legible por ordenador que transporta o comunica un código de programa en forma de instrucciones o estructuras de datos, tales como señales u ondas propagadas, a las que se puede acceder mediante un ordenador, que las puede leer y/o ejecutar.

40 [0178] El código de programa se puede ejecutar mediante un procesador, que puede incluir uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Dicho procesador puede estar configurado para realizar cualquiera de las técnicas descritas en esta divulgación. Un procesador de propósito general puede ser un microprocesador, pero, de forma alternativa el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también se puede implementar como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo. En consecuencia, el término "procesador", como se usa en el presente documento se puede referir a cualquiera de las estructuras anteriores, cualquier combinación de las estructuras anteriores, o cualquier otra estructura o aparato adecuados para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de software o módulos de hardware dedicados configurados para codificar y descodificar, o incorporados en un codificador-descodificador de vídeo combinado (CÓDEC). Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

50 [0179] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

65

[0180] Se han descrito diversos modos de realización de la invención. Estos y otros modos de realización están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un aparato configurado para codificar información de vídeo, comprendiendo el aparato:

5 una unidad de memoria configurada para almacenar información de vídeo asociada a una capa de referencia (RL) y una capa de mejora (EL), comprendiendo la EL una unidad de vídeo EL y comprendiendo la RL una unidad de vídeo RL correspondiente a la unidad de vídeo EL; y

10 un procesador en comunicación con la unidad de memoria, estando configurado el procesador para:

realizar un sobremuestreo espacial y una conversión de profundidad de bits en una información de píxeles de la unidad de vídeo RL en un solo proceso para determinar una información de píxeles predichos de la unidad de vídeo EL, en el que el proceso único comprende un conjunto de operaciones realizadas en conexión con el sobremuestreo espacial, en el que una de las operaciones comprende un proceso de sobremuestreo en el que un desplazamiento a la derecha de un desplazamiento de bits a la derecha se reduce en una diferencia de profundidad de bits entre una profundidad de bits de la BL y una profundidad de bits de la EL para lograr también la conversión de profundidad de bits en el que el desplazamiento de bits a la derecha es de T-K bits donde T es un parámetro de desplazamiento de bits a la izquierda del proceso de sobremuestreo y K es la diferencia de profundidad de bits entre la capa base y la capa de mejora; y determinar la unidad de vídeo EL usando la información de píxeles predichos.

2. El aparato de la reivindicación 1, en el que la unidad de vídeo EL es una de entre una imagen EL y un bloque EL dentro de la imagen EL, y la unidad de vídeo RL es una de entre una imagen RL y un bloque RL dentro de la imagen RL.

3. El aparato de la reivindicación 1, en el que la conversión de profundidad de bits hace que la unidad de vídeo RL sobremuestreada tenga el mismo número de bits que se usa para la unidad de vídeo EL.

4. El aparato de la reivindicación 1, en el que el procesador está configurado para aplicar un filtro combinado de sobremuestreo y conversión de profundidad de bits a la información de píxeles de la unidad de vídeo RL, estando configurado el filtro combinado de sobremuestreo y conversión de profundidad de bits para sobremuestrear la información de píxeles de la unidad de vídeo RL en base una relación de resolución de la RL y la EL y para convertir la profundidad de bits de la información de píxeles sobremuestreada en base a la diferencia entre la profundidad de bits de la EL y la profundidad de bits de la RL.

5. El aparato de la reivindicación 1, en el que el procesador está configurado para aplicar un filtro combinado de conversión de profundidad de bits y sobremuestreo a la información de píxeles de la unidad de vídeo RL, estando configurado el filtro combinado de conversión de profundidad de bits y sobremuestreo para convertir la profundidad de bits de la información de píxeles de la unidad de vídeo RL en base a la diferencia entre la profundidad de bits de la EL y la profundidad de bits de la RL, y sobremuestrear la información de píxeles convertida en base a una relación de resolución de la RL y la EL.

6. El aparato de la reivindicación 1, en el que el procesador está configurado para aplicar un filtro de sobremuestreo y conversión de profundidad de bits que tiene uno o más factores de peso, un desplazamiento, y un valor de desplazamiento de bits,

en el que el valor de desplazamiento de bits depende de la diferencia entre una profundidad de bits EL asociada a la EL y una profundidad de bits RL asociada a la RL.

7. El aparato de la reivindicación 1, en el que el aparato comprende un codificador, y en el que el procesador está configurado además para codificar la unidad de vídeo EL usando la información de píxeles predichos.

8. El aparato de la reivindicación 1, en el que el aparato comprende un descodificador, y en el que el procesador está configurado además para descodificar la unidad de vídeo EL usando la información de píxeles predichos.

9. Un procedimiento de codificación de información de vídeo, comprendiendo el procedimiento:

60 almacenar información de vídeo asociada a una capa de referencia (RL) y una capa de mejora (EL), comprendiendo la EL una unidad de vídeo EL y comprendiendo la RL una unidad de vídeo RL correspondiente a la unidad de vídeo EL;

realizar un sobremuestreo espacial y una conversión de profundidad de bits en la información de píxeles de la unidad de vídeo RL en un solo proceso para determinar una información de píxeles predichos de la unidad de vídeo EL, en el que el proceso único comprende un conjunto de operaciones realizadas en conexión con el sobremuestreo espacial, en el que una de las operaciones comprende un proceso de sobremuestreo en el que un desplazamiento a la derecha de un desplazamiento de bits a la derecha se reduce en una diferencia

de profundidad de bits entre una profundidad de bits de la BL y una profundidad de bits de la EL para lograr también la conversión de profundidad de bits en el que el desplazamiento de bits a la derecha es de $T-K$ bits donde T es un parámetro de desplazamiento de bits a la izquierda del proceso de sobremuestreo y K es la diferencia de profundidad de bits entre la capa base y la capa de mejora; y

5

determinar la unidad de vídeo EL usando la información de píxeles predichos.

10 **10.** El procedimiento de la reivindicación 9, en el que la unidad de vídeo EL es una de entre una imagen EL y un bloque EL dentro de la imagen EL, y la unidad de vídeo RL es una de entre una imagen RL y un bloque RL dentro de la imagen RL.

11. El procedimiento de la reivindicación 9, en el que la conversión de profundidad de bits hace que la unidad de vídeo RL sobremuestreada tenga el mismo número de bits que se usa para la unidad de vídeo EL.

15 **12.** Un medio no transitorio legible por ordenador que comprende un código que, cuando se ejecuta, hace que un aparato realice el procedimiento de cualquiera de las reivindicaciones 9, 10 u 11.

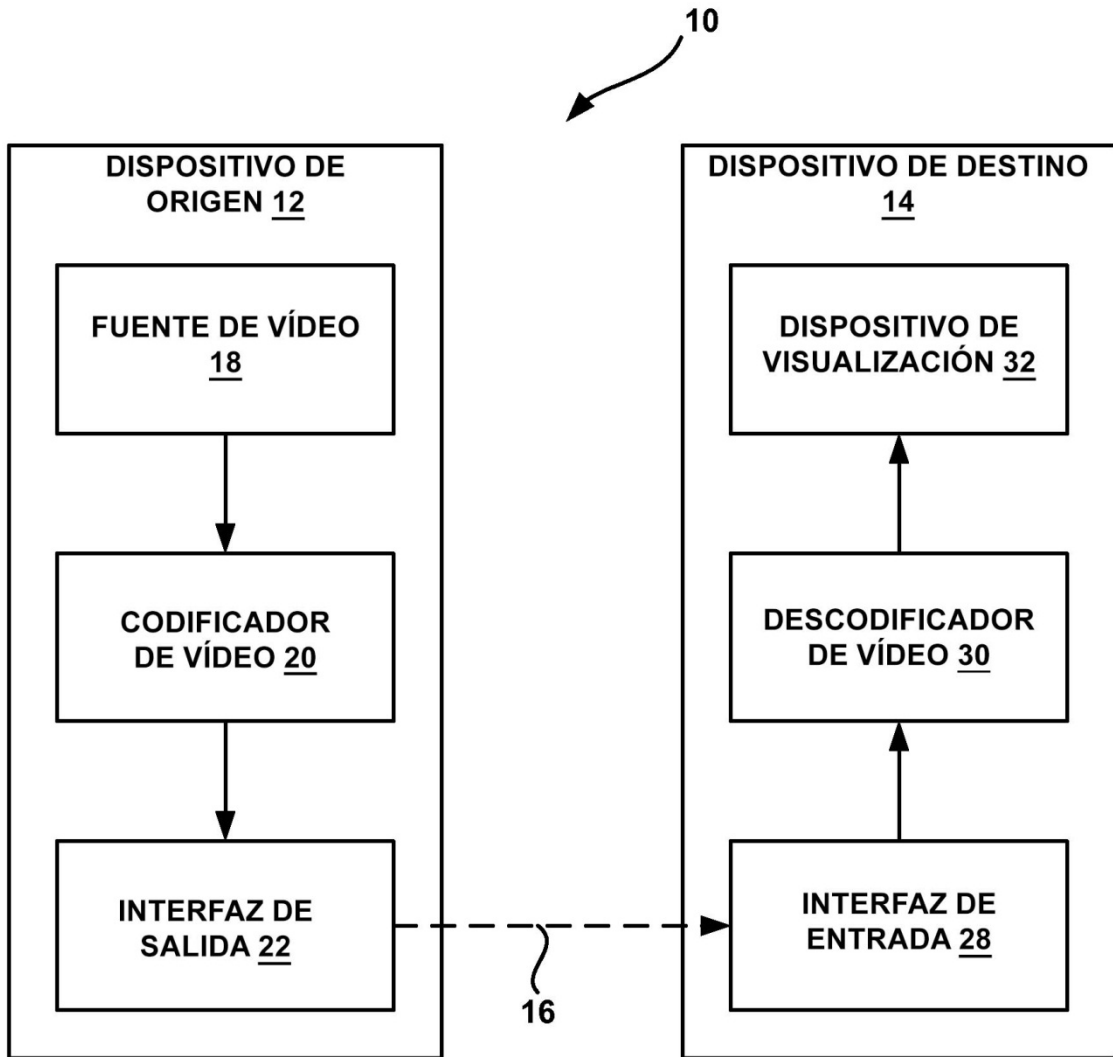


FIG. 1

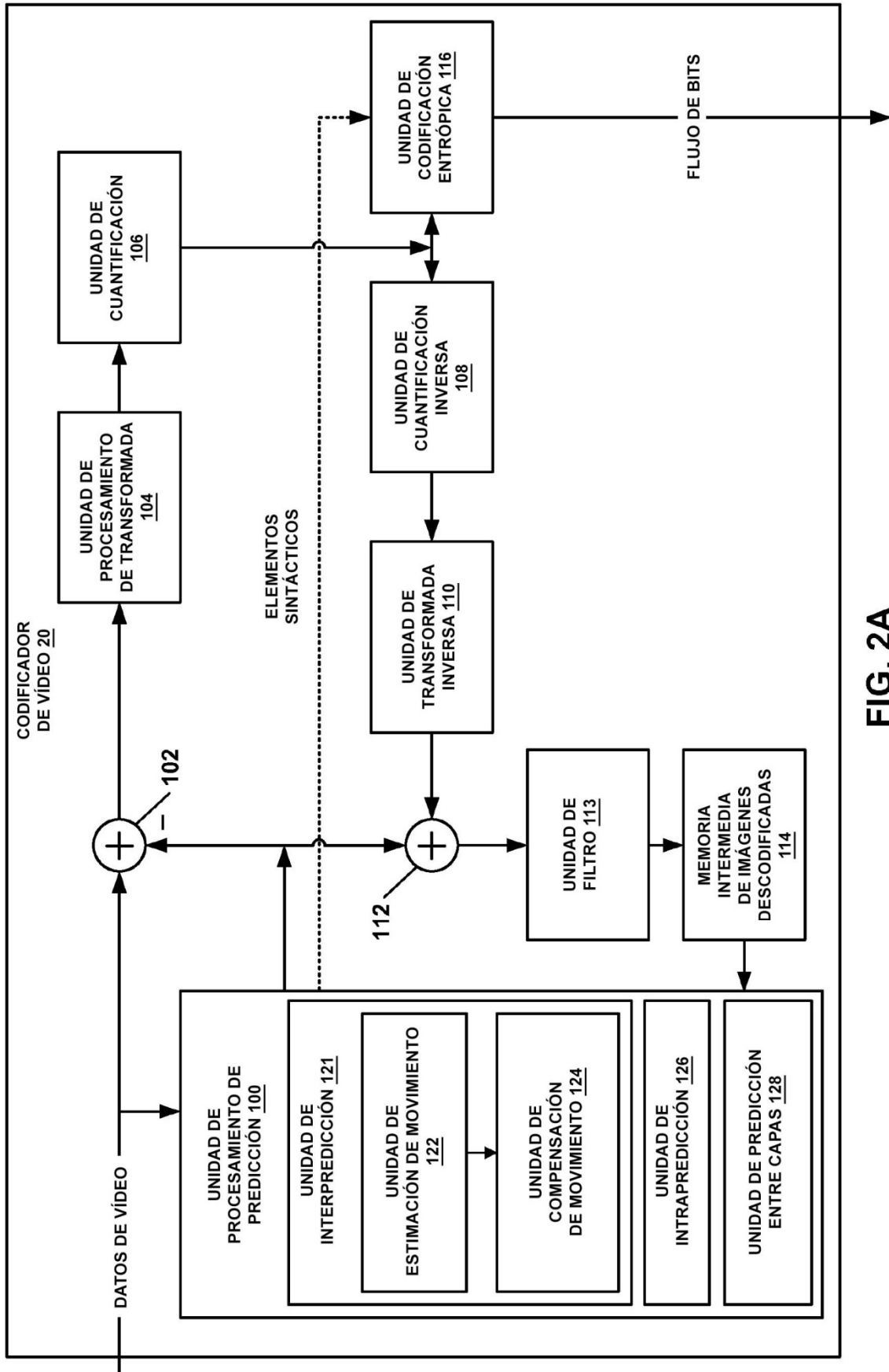


FIG. 2A

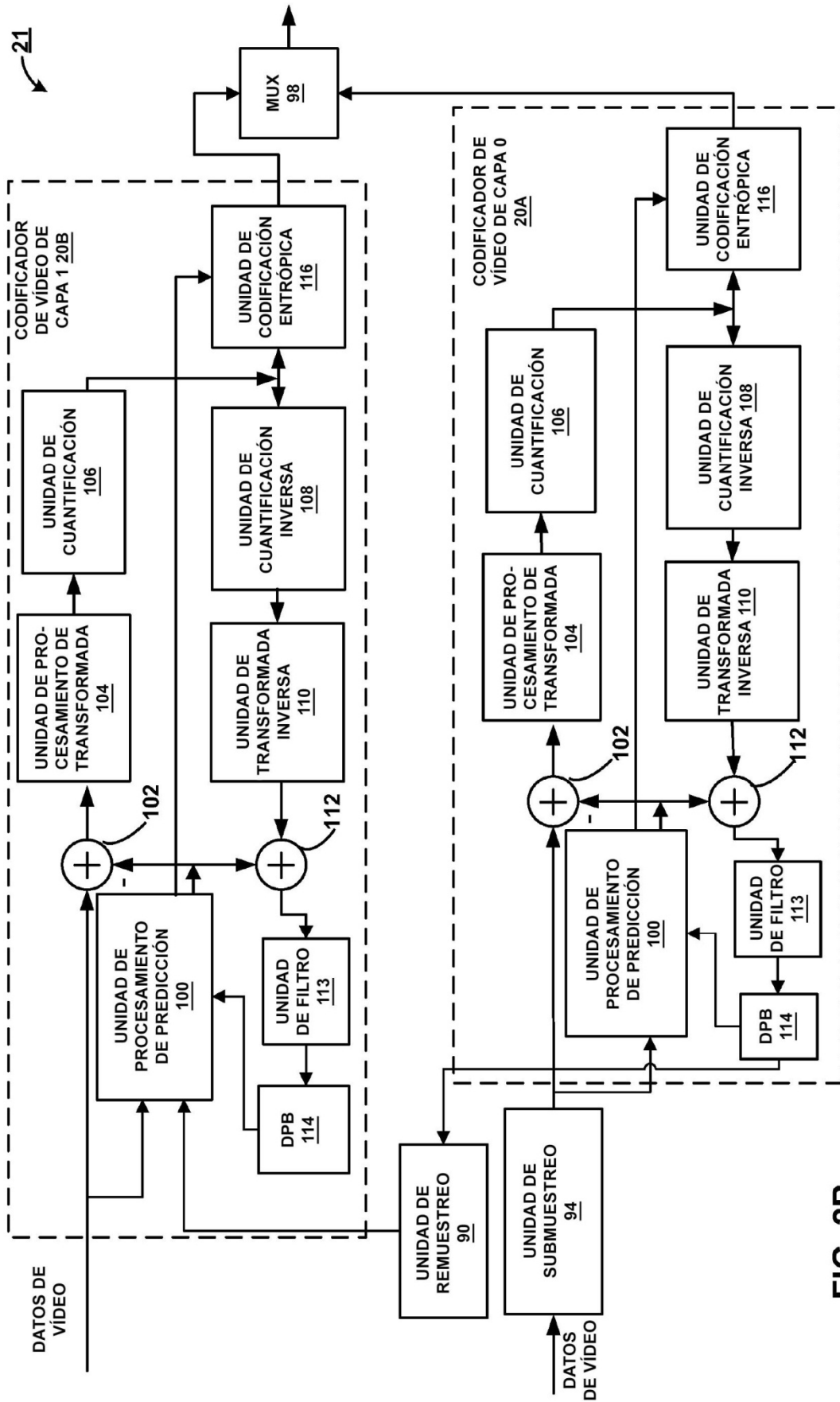


FIG. 2B

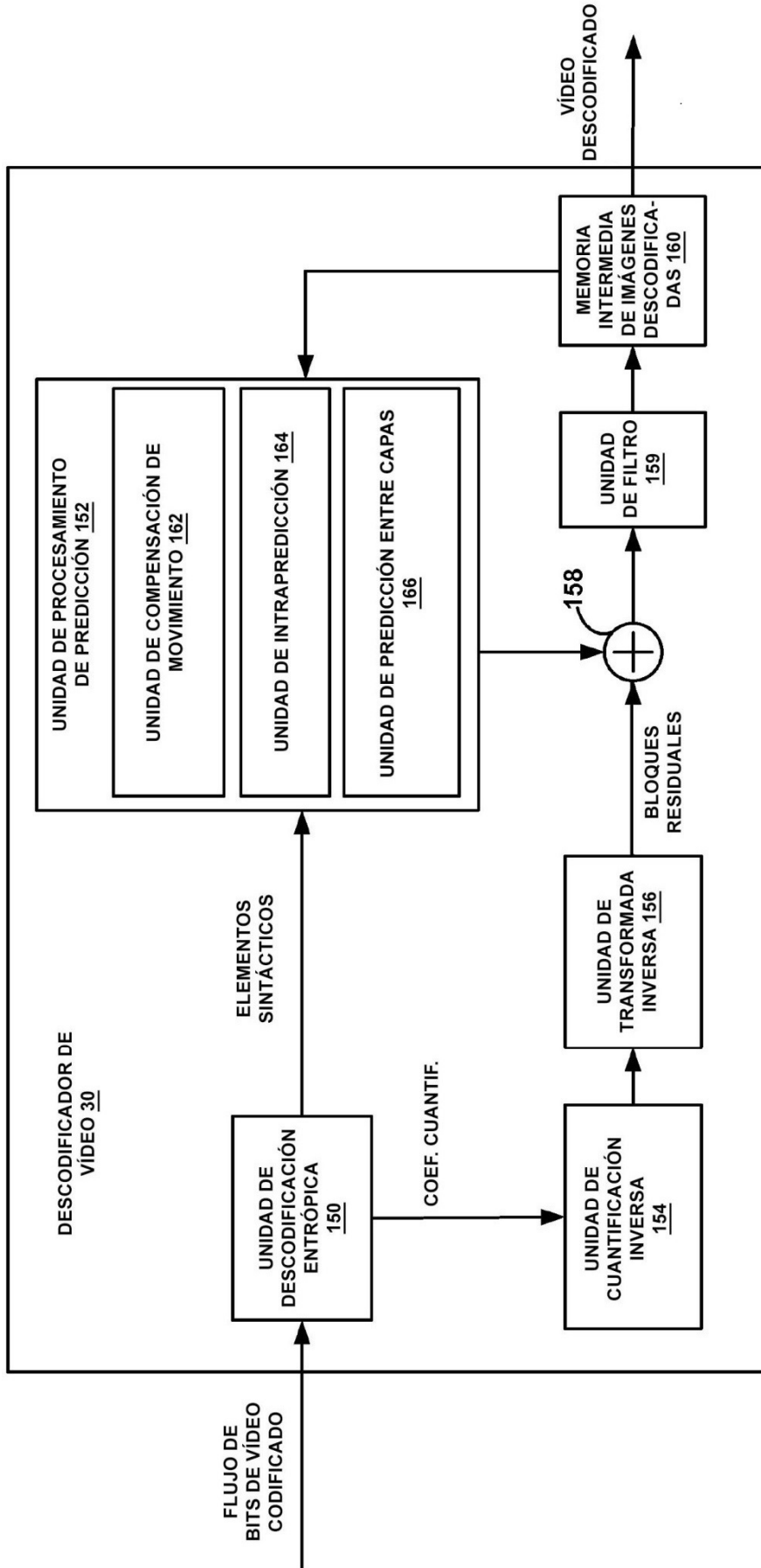


FIG. 3A

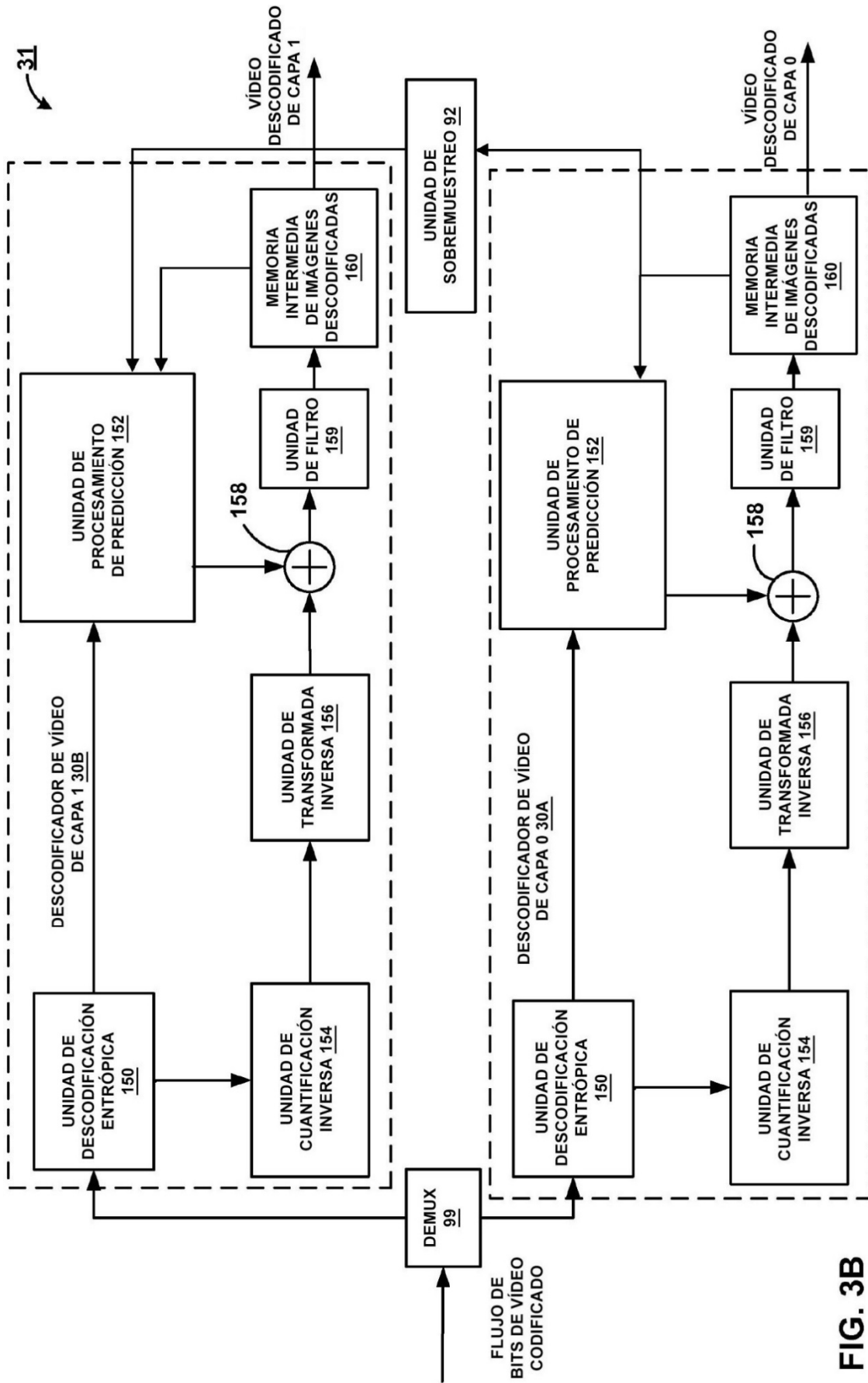


FIG. 3B

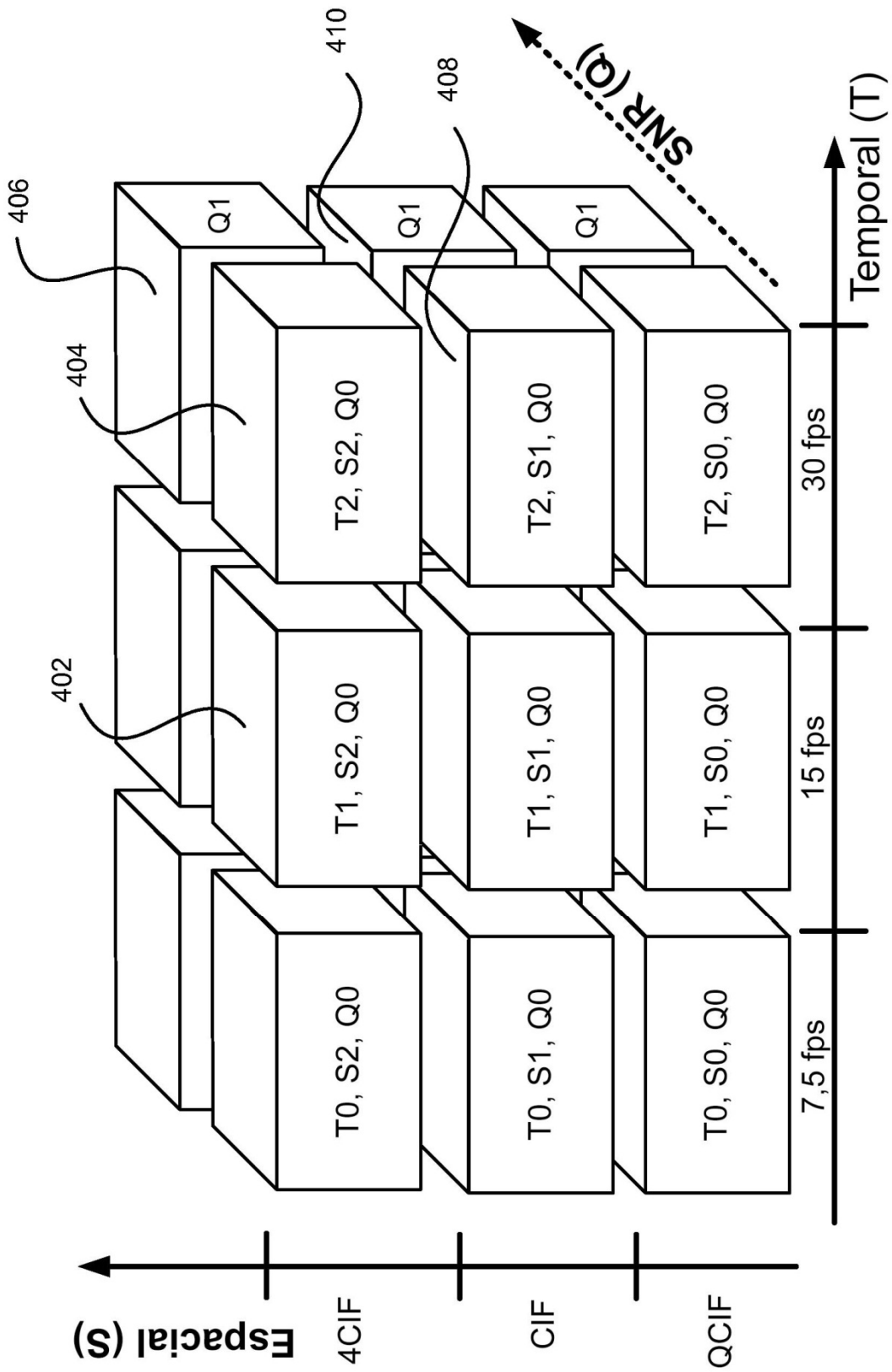


FIG. 4

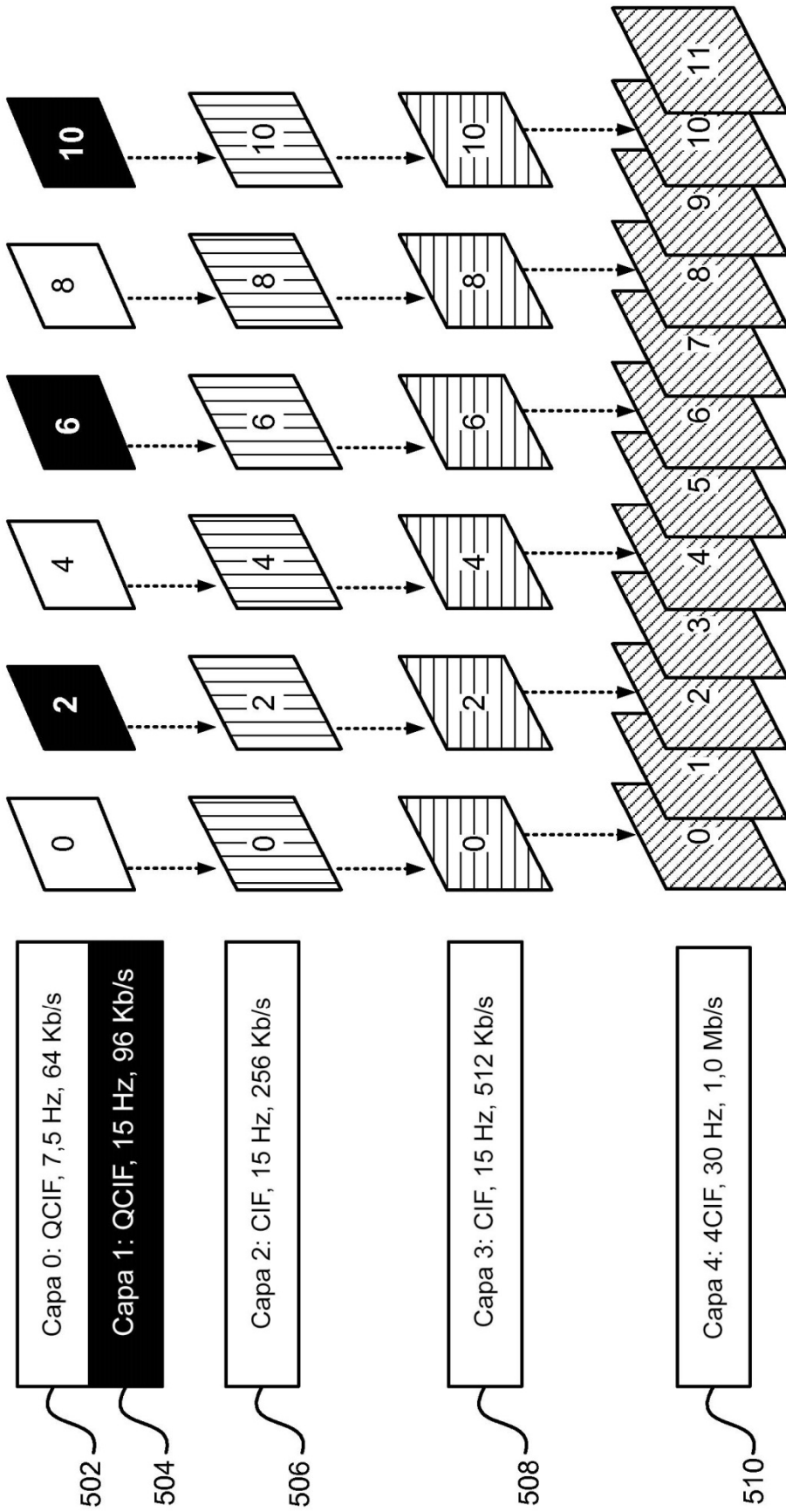


FIG. 5

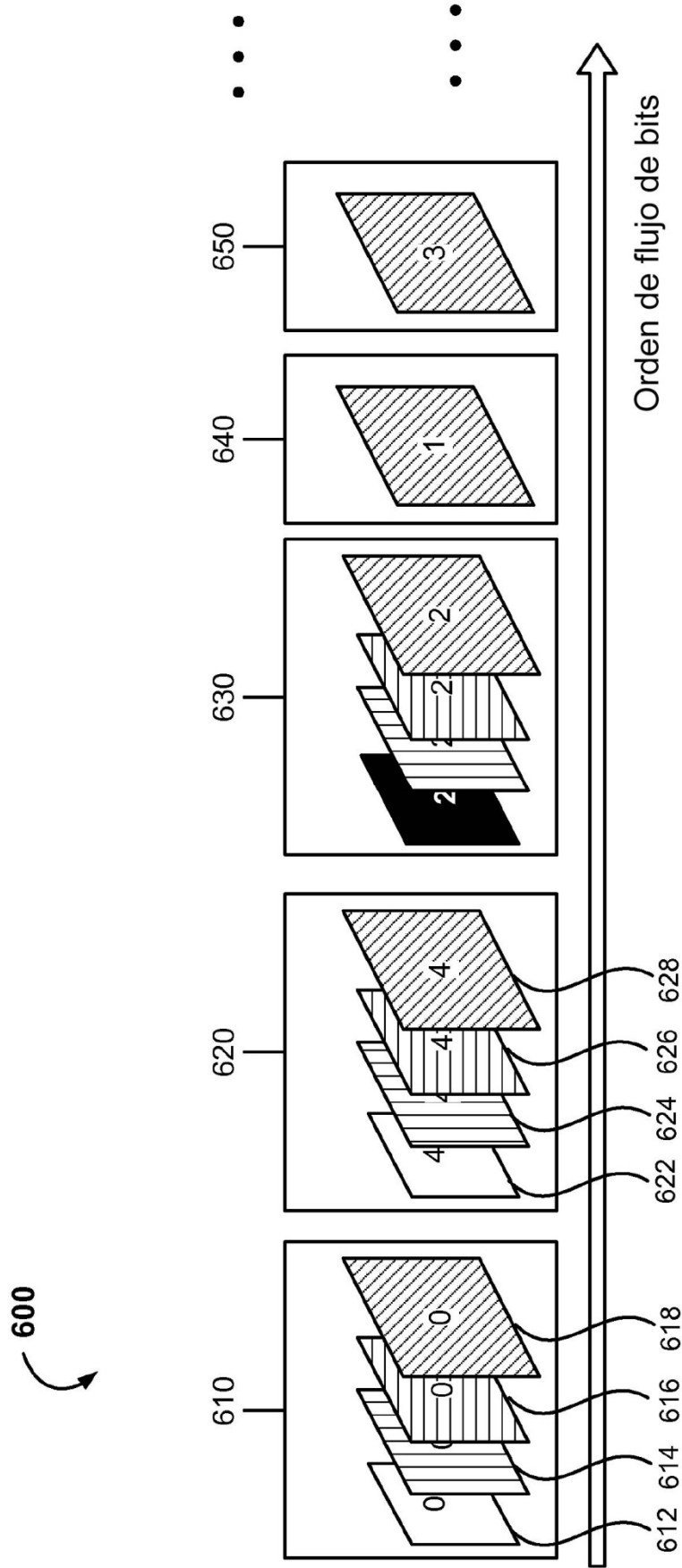


FIG. 6

700

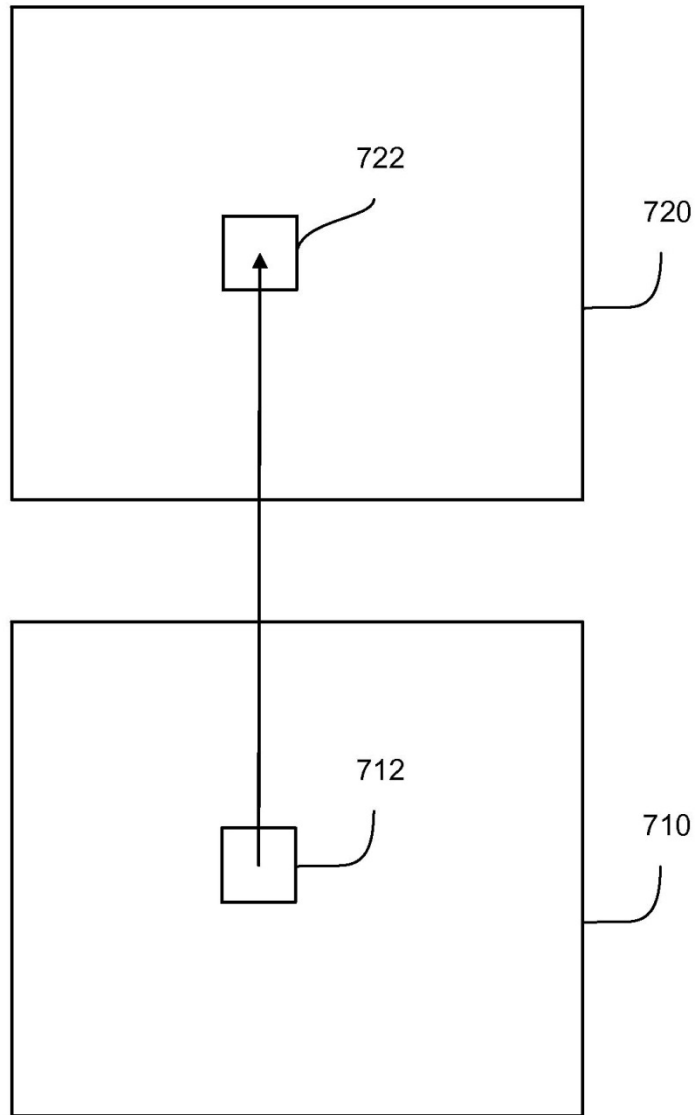



FIG. 7

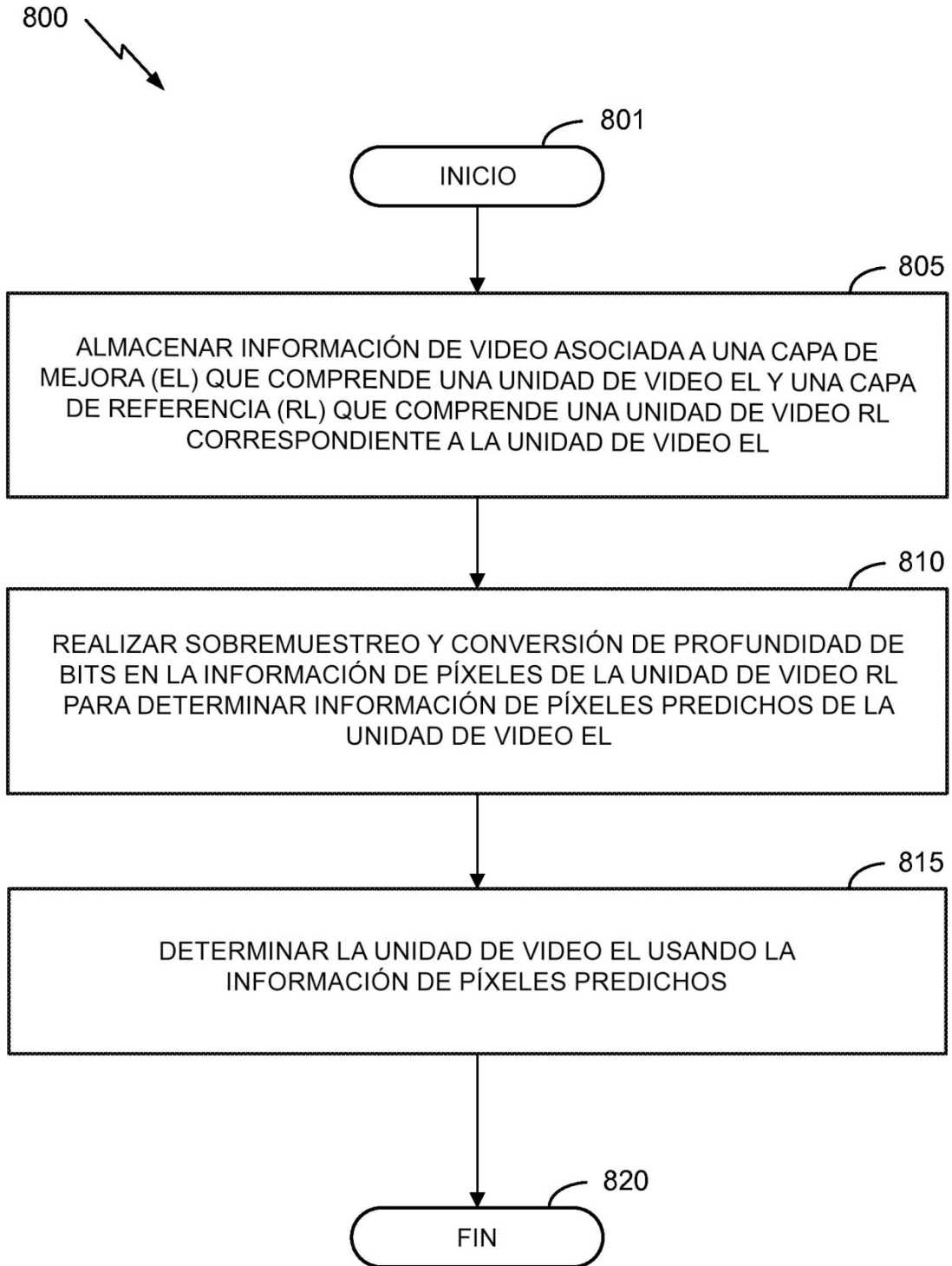


FIG. 8

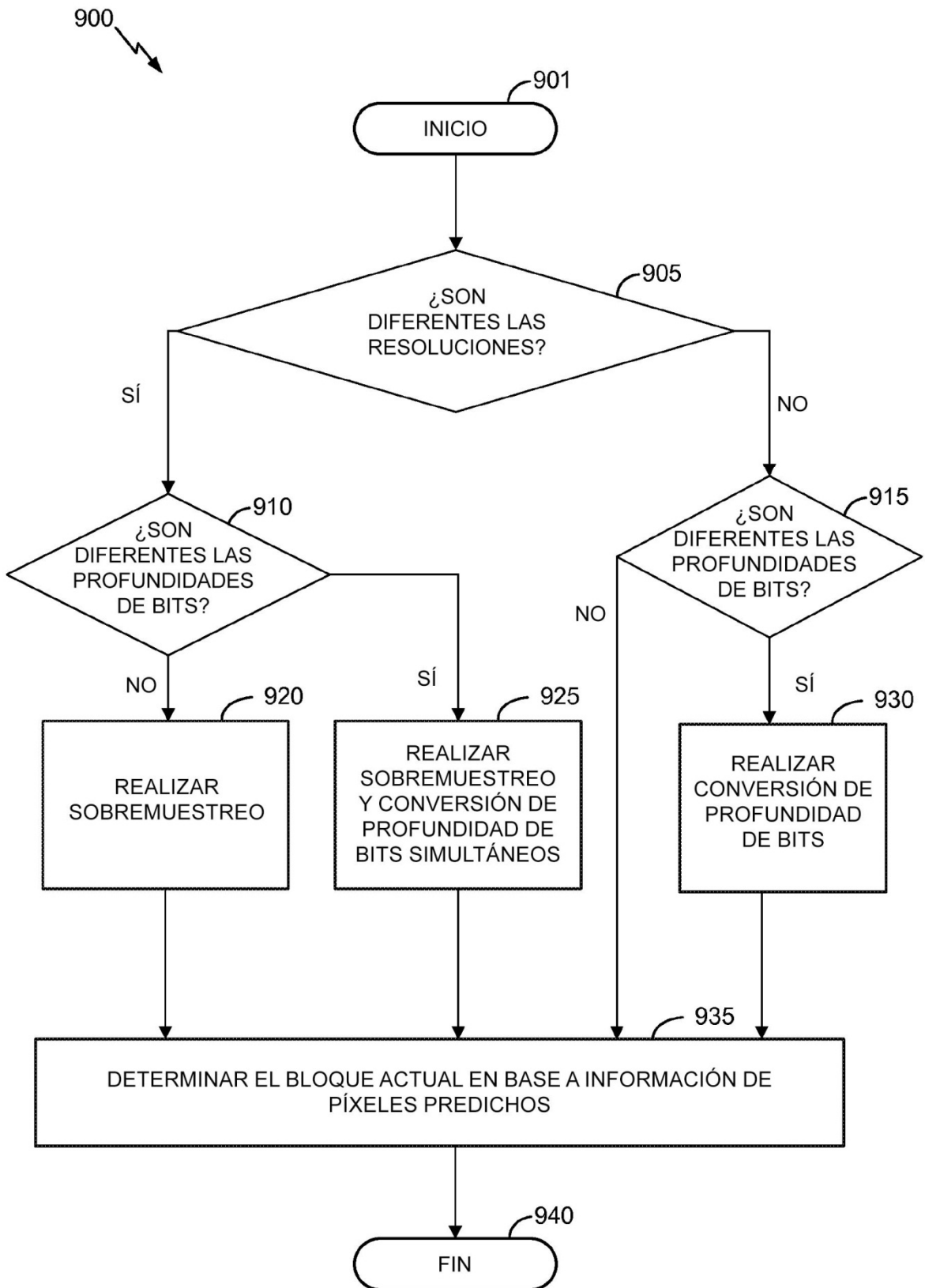


FIG. 9