

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 013 317**

51 Int. Cl.:

**H04N 19/82** (2014.01)

**H04N 19/31** (2014.01)

**H04N 19/463** (2014.01)

**H04N 19/117** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **04.01.2018 PCT/US2018/012355**

87 Fecha y número de publicación internacional: **12.07.2018 WO18129168**

96 Fecha de presentación y número de la solicitud europea: **04.01.2018 E 18701627 (4)**

97 Fecha y número de publicación de la concesión europea: **19.02.2025 EP 3566444**

54 Título: **Predicción temporal de filtro de bucle adaptativo modificado para soporte de escalabilidad temporal**

30 Prioridad:

**04.01.2017 US 201762442322 P**

**11.01.2017 US 201762445174 P**

**03.01.2018 US 201815861165**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.04.2025**

73 Titular/es:

**QUALCOMM INCORPORATED (100.00%)**  
**ATTN: International IP Administration 5775**  
**Morehouse Drive**  
**San Diego, CA 92121-1714, US**

72 Inventor/es:

**ZHANG, LI;**  
**CHIEN, WEI-JUNG;**  
**KARCZEWICZ, MARTA y**  
**WANG, YE-KUI**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 3 013 317 T3

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

## DESCRIPCIÓN

Predicción temporal de filtro de bucle adaptativo modificado para soporte de escalabilidad temporal

5 Esta solicitud reivindica el beneficio de la Solicitud Provisional de los Estados Unidos No. 62/442,322, presentada el 4 de enero de 2017, y la Solicitud Provisional de los Estados Unidos No. 62/445,174, presentada el 11 de enero de 2017.

Campo técnico

10 Esta divulgación se relaciona con la codificación y decodificación de vídeo.

Antecedentes

15 Las capacidades de vídeo digital se pueden incorporar en un amplio rango de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores tipo tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio satelital, los llamados  
20 "teléfonos inteligentes", dispositivos de teleconferencia por vídeo, dispositivos de generación de flujo de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), el estándar ITU-T H.265, de Codificación de Vídeo de Alta Eficiencia (HEVC) y extensiones de tales estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar,  
25 decodificar y/o almacenar información de vídeo digital de manera más eficiente al implementar tales técnicas de compresión de vídeo.

30 Las técnicas de compresión de vídeo pueden realizar predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o retirar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una sección de vídeo (por ejemplo, un marco de vídeo o una porción de un marco de vídeo) puede dividirse en bloques de vídeo, tales como bloques de árbol de codificación y bloques de codificación. La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que va a ser codificado. Los datos residuales representan diferencias de píxeles entre el bloque original que va a ser codificado y el bloque predictivo. Para una compresión adicional, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformada, dando como resultado coeficientes de transformada  
35 residuales, que luego pueden cuantificarse.

40 JVET-E0030 de Sjöberg et al, 2 de enero de 2017, describe "JEM bug fix for ALF cross-picture prediction" que reemplaza un arreglo de ALF separado que almacenaba parámetros de ALF de como máximo seis imágenes previas con una lista que solo contiene indicadores de imágenes que se usan para la predicción a través de los mecanismos de RPS con el fin de evitar que los parámetros de ALF a veces se copien de imágenes de referencia de capas temporales superiores a la imagen actual.

45 El documento US 2013/022104 describe conjuntos de parámetros de Adaptación para codificación de vídeo y técnicas de gestión de memoria para determinar que los conjuntos de parámetros de adaptación se pueden retirar de un búfer de un dispositivo de codificación de vídeo para evitar almacenar datos en exceso en un búfer. Se puede predecir la totalidad o una porción de un APS a partir de uno o más APS previos. El APS puede incluir una indicación de que sus coeficientes de ALF son los mismos que un primer APS previo o los coeficientes de ALF se pueden señalar explícitamente. Un APS asociado con una imagen codificada se puede  
50 restringir para que haga referencia solamente a un APS asociado con imágenes codificadas que tengan un valor temporal\_ID igual o inferior, de tal manera que la extracción de subflujo de bits no impida el uso del APS dependiente.

Resumen

55 La invención se define en las reivindicaciones independientes anexas y corresponde a la primera y segunda técnicas que se discuten a continuación. Las técnicas restantes divulgadas a continuación, tomadas en aislamiento de cualquiera de la primera y segunda técnicas, deben entenderse como ejemplos ilustrativos no cubiertos por la invención reivindicada.

60 Las características opcionales se definen en las reivindicaciones dependientes. En general, esta divulgación describe técnicas relacionadas con el filtrado de bucle adaptativo (ALF), especialmente para la predicción de filtros de ALF a partir de marcos, secciones o mosaicos codificados previamente. Las técnicas se pueden usar en el contexto de códecs de vídeo avanzados, tales como extensiones de HEVC o la próxima generación de  
65 estándares de codificación de vídeo.

Los detalles de uno o más aspectos de la divulgación se establecen en los dibujos acompañantes y en la descripción a continuación. Otras características, objetos y ventajas de las técnicas descritas en esta divulgación serán evidentes a partir de la descripción, dibujos y reivindicaciones.

- 5 Breve descripción de los dibujos  
  
La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede usar una o más técnicas descritas en esta divulgación.
- 10 La figura 2 ilustra tres diferentes soportes de filtro de Filtrado de Bucle Adaptativo (ALF) de ejemplo.  
  
La figura 3 ilustra un ejemplo de configuración de Acceso Aleatorio con un Grupo de Imágenes (GOP) igual a 16.
- 15 La figura 4A ilustra un arreglo para almacenar parámetros de filtro.  
  
La figura 4B ilustra un estado diferente del arreglo para almacenar parámetros de filtro.
- 20 La figura 5 ilustra una pluralidad de arreglos que corresponden a diferentes capas temporales, de acuerdo con una primera técnica de esta divulgación.  
  
La figura 6 ilustra un arreglo para almacenar parámetros de ALF y valores de índice de capa temporal asociados, de acuerdo con una segunda técnica de esta divulgación.
- 25 La figura 7 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar una o más técnicas descritas en esta divulgación.  
  
La figura 8 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar una o más técnicas descritas en esta divulgación.
- 30 La figura 9 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una primera técnica de esta divulgación.  
  
La figura 10 es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de vídeo, de acuerdo con la primera técnica de esta divulgación.
- 35 La figura 11 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una segunda técnica de esta divulgación.  
  
La figura 12 es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de vídeo, de acuerdo con la segunda técnica de esta divulgación.
- Descripción detallada
- 45 El Filtrado de Bucle Adaptativo (ALF) es un proceso que aplica uno o más filtros adaptativos (es decir, unos filtros de ALF) como parte de un bucle de codificación para mejorar la calidad de los datos de vídeo decodificados. Un filtro de ALF está asociado con un conjunto de coeficientes. Un codificador de vídeo (es decir, un codificador de vídeo o un decodificador de vídeo) puede aplicar filtros de ALF con diferentes coeficientes a diferentes bloques de la misma imagen, con base en las características de los bloques. Para
- 50 reducir la sobrecarga asociada con la señalización de los coeficientes asociados con los filtros de ALF, un codificador de vídeo puede almacenar, en un arreglo, conjuntos de parámetros de ALF para filtros de ALF usados en imágenes, mosaicos o secciones codificados previamente. Un conjunto de parámetros de ALF puede incluir múltiples coeficientes asociados con uno o más filtros de ALF. Por ejemplo, un conjunto de parámetros de ALF puede indicar coeficientes asociados con múltiples filtros. El codificador de vídeo reemplaza conjuntos de parámetros de ALF en el arreglo de acuerdo con una base de Primero en Entrar Primero en Salir (FIFO).
- 55
- 60 Diferentes imágenes en una secuencia de vídeo pueden pertenecer a diferentes capas temporales. Diferentes capas temporales están asociadas con diferentes identificadores temporales. Una imagen en una capa temporal dada puede ser decodificada con referencia a otras imágenes que tengan el identificador temporal de la capa temporal dada e imágenes que tengan identificadores temporales con valores menores que los valores del identificador temporal de la capa temporal dada.
- 65 Debido a que un codificador de vídeo almacena datos de filtro (por ejemplo, conjuntos de parámetros de ALF) en el arreglo de acuerdo con una base de FIFO, el arreglo puede contener datos de filtro de una imagen que tenga un identificador temporal superior que un identificador temporal de una imagen que se decodifica actualmente. Esto podría causar potencialmente errores en el proceso de filtrado debido a que esto puede

hacer que la imagen actual dependa de una imagen en una capa temporal con un identificador temporal superior que la capa temporal de la imagen actual si la imagen con el identificador temporal superior se pierde o no es necesario que se decodifique.

Esta divulgación describe técnicas que pueden abordar esta deficiencia. En un ejemplo, un codificador de vídeo puede almacenar, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de uno o más filtros de ALF a muestras de regiones de imágenes de los datos de vídeo codificados antes de la imagen actual. Cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Además, el codificador de vídeo puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo que corresponde a una capa temporal a la cual pertenece una región actual, un conjunto aplicable de parámetros de ALF para la región actual. Esta divulgación puede usar el término "región" para referirse a una sección u otro tipo de área de una imagen actual para realizar ALF. El codificador de vídeo puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, un filtro de ALF a la región actual.

En algunos ejemplos, un codificador de vídeo almacena, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de uno o más filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Adicionalmente, en este ejemplo, el codificador de vídeo almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF. Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. En este ejemplo, el codificador de vídeo puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. Además, en este ejemplo, el codificador de vídeo puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual.

En cualquiera de estos ejemplos, asociar parámetros de ALF con capas temporales puede ayudar a evitar el problema de que una imagen actual dependa potencialmente de la decodificación de una imagen en una capa temporal superior.

La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede usar técnicas de esta divulgación. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados para ser decodificados en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo codificados al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de un amplio rango de dispositivos, incluyendo ordenadores de escritorio, ordenadores portables (es decir, portátiles), ordenadores tipo tableta, decodificadores, aparatos telefónicos tales como los denominados teléfonos "inteligentes", ordenadores tipo tableta, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de generación de flujo de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 están equipados para la comunicación inalámbrica. De este modo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden ser dispositivos de comunicación inalámbrica. Las técnicas descritas en esta divulgación se pueden aplicar a aplicaciones inalámbricas y/o cableadas. El dispositivo de origen 12 es un dispositivo de codificación de vídeo de ejemplo (es decir, un dispositivo para codificar datos de vídeo). El dispositivo de destino 14 es un dispositivo de decodificación de vídeo de ejemplo (es decir, un dispositivo para decodificar datos de vídeo).

El sistema 10 ilustrado en la figura 1 es simplemente un ejemplo. Las técnicas para procesar datos de vídeo pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. En algunos ejemplos, las técnicas pueden ser realizadas por un codificador de vídeo, típicamente denominado como un "CODEC". El dispositivo de origen 12 y el dispositivo de destino 14 son ejemplos de tales dispositivos de codificación en los cuales el dispositivo de origen 12 genera datos de vídeo codificados para transmisión al dispositivo de destino 14. En algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 operan de una manera sustancialmente simétrica de tal manera que cada uno del dispositivo de origen 12 y el dispositivo de destino 14 incluye componentes de codificación y decodificación de vídeo. Por tanto, el sistema 10 puede soportar la transmisión de vídeo unidireccional o bidireccional entre el dispositivo de origen 12 y el dispositivo de destino 14, por ejemplo, para generación de flujo de vídeo, reproducción de vídeo, radiodifusión de vídeo o telefonía por vídeo.

En el ejemplo de la figura 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, medio de almacenamiento 19 configurado para almacenar datos de vídeo, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 26, medio de almacenamiento 28 configurado para almacenar datos de vídeo codificados, un decodificador de vídeo 30 y dispositivo de visualización 32. En otros ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 incluyen otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo de una fuente de vídeo externa, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede

interconectarse con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

La fuente de vídeo 18 es una fuente de datos de vídeo. Los datos de vídeo pueden comprender una serie de imágenes. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente y/o una interfaz de transmisión de vídeo para recibir datos de vídeo de un proveedor de contenido de vídeo. En algunos ejemplos, la fuente de vídeo 18 genera datos de vídeo basados en gráficos de ordenador o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. Los medios de almacenamiento 19 pueden estar configurados para almacenar los datos de vídeo. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede estar codificado por el codificador de vídeo 20.

La interfaz de salida 22 puede emitir la información de vídeo codificada a un medio legible por ordenador 16. La interfaz de salida 22 puede comprender diversos tipos de componentes o dispositivos. Por ejemplo, la interfaz de salida 22 puede comprender un transmisor inalámbrico, un módem, un componente de red cableada (por ejemplo, una tarjeta Ethernet) u otro componente físico. En ejemplos donde la interfaz de salida 22 comprende un transmisor inalámbrico, la interfaz de salida 22 puede estar configurada para transmitir datos, tales como datos de vídeo codificados, modulados de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE, LTE Avanzada, 5G y similares. En algunos ejemplos donde la interfaz de salida 22 comprende un transmisor inalámbrico, la interfaz de salida 22 puede estar configurada para transmitir datos, tales como datos de vídeo codificados, modulados de acuerdo con otros estándares inalámbricos, tales como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ y similares. De este modo, en algunos ejemplos, el dispositivo de origen 12 comprende un dispositivo de comunicación inalámbrica que incluye un transmisor configurado para transmitir datos de vídeo codificados. En algunos de tales ejemplos, el dispositivo de comunicación inalámbrica comprende un aparato telefónico y el transmisor está configurado para modular, de acuerdo con un estándar de comunicación inalámbrica, una señal que comprende los datos de vídeo codificados.

En algunos ejemplos, la circuitería de interfaz de salida 22 está integrada en la circuitería de codificador de vídeo 20 y/o en otros componentes del dispositivo de origen 12. Por ejemplo, el codificador de vídeo 20 y la interfaz de salida 22 pueden ser partes de un sistema en un chip (SoC). El SoC también puede incluir otros componentes, tales como un microprocesador de propósito general, una unidad de procesamiento de gráficos, y así sucesivamente.

El dispositivo de destino 14 puede recibir datos de vídeo codificados para ser decodificados a través de un medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En algunos ejemplos, el medio legible por ordenador 16 comprende un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión 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 el Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14. El dispositivo de destino 14 puede comprender uno o más medios de almacenamiento de datos configurados para almacenar datos de vídeo codificados y datos de vídeo decodificados.

En algunos ejemplos, la interfaz de salida 22 puede emitir datos, tales como datos de vídeo codificados, a un dispositivo intermedio, tal como un dispositivo de almacenamiento. De manera similar, la interfaz de entrada 26 del dispositivo de destino 14 puede recibir datos codificados del dispositivo intermedio. El dispositivo intermedio puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o accedidos localmente tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En algunos ejemplos, el dispositivo intermedio corresponde a un servidor de archivos. Los servidores de archivos de ejemplo incluyen servidores web, servidores de FTP, dispositivos de almacenamiento conectados a red (NAS) o unidades de disquete locales.

El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem por cable, etc.) o una combinación de ambas que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de los datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión de generación de flujo, una transmisión de descarga o una combinación de ambas.

El medio legible por ordenador 16 puede incluir medios transitorios, tal como una radiodifusión inalámbrica o transmisión por red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no

transitorios), tales como un disquete duro, unidad flash, disco compacto, disco de vídeo digital, disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no se muestra) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, a través de una transmisión por red. De manera similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz de entrada 26 del dispositivo de destino 14 recibe datos desde un medio legible por ordenador 16. La interfaz de entrada 26 puede comprender diversos tipos de componentes o dispositivos. Por ejemplo, la interfaz de entrada 26 puede comprender un receptor inalámbrico, un módem, un componente de red cableada (por ejemplo, una tarjeta Ethernet) u otro componente físico. En los ejemplos donde la interfaz de entrada 26 comprende un receptor inalámbrico, la interfaz de entrada 26 puede estar configurada para recibir datos, tal como el flujo de bits, modulados de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE, LTE Avanzada, 5G y similares. En algunos ejemplos donde la interfaz de entrada 26 comprende un receptor inalámbrico, la interfaz de entrada 26 puede estar configurada para recibir datos, tal como el flujo de bits, modulados de acuerdo con otros estándares inalámbricos, tales como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ y similares. De este modo, en algunos ejemplos, el dispositivo de destino 14 puede comprender un dispositivo de comunicación inalámbrica que comprende un receptor configurado para recibir datos de vídeo codificados. En algunos de tales ejemplos, el dispositivo de comunicación inalámbrica comprende un aparato telefónico y el receptor está configurado para desmodular, de acuerdo con un estándar de comunicación inalámbrica, una señal que comprende los datos de vídeo codificados. En algunos ejemplos, el dispositivo de origen 12 puede comprender un transmisor y los dispositivos de destino 14 pueden comprender un transmisor y receptor.

En algunos ejemplos, la circuitería de la interfaz de entrada 26 puede estar integrada en la circuitería del decodificador de vídeo 30 y/o en otros componentes del dispositivo de destino 14. Por ejemplo, el decodificador de vídeo 30 y la interfaz de entrada 26 pueden ser partes de un SoC. El SoC también puede incluir otros componentes, tales como un microprocesador de propósito general, una unidad de procesamiento de gráficos, y así sucesivamente.

El medio de almacenamiento 28 puede estar configurado para almacenar datos de vídeo codificados, tales como datos de vídeo codificados (por ejemplo, un flujo de bits) recibidos por la interfaz de entrada 26. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario. El dispositivo de visualización 32 puede comprender cualquiera de una variedad de dispositivos de visualización tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánico (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitería programable y/o de función fija adecuada, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), arreglos de puertas programables en campo (FPGA), lógica discreta, software, hardware, firmware 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 legible por ordenador no transitorio, adecuado y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y del decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 codifican y decodifican datos de vídeo de acuerdo con un estándar o especificación de codificación de vídeo. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar y decodificar datos de vídeo de acuerdo con 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 conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de Codificación de Vídeo Escalable (SVC) y Codificación de Vídeo MultiVista (MVC), u otro estándar o especificación de codificación de vídeo. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 codifican y decodifican datos de vídeo de acuerdo con la Codificación de Vídeo de Alta Eficiencia (HEVC), que se conoce como o ITU-T H.265, sus extensiones de codificación de contenido de pantalla y rango, su extensión de codificación de vídeo 3D (3D-HEVC), su extensión multivista (MV-HEVC) o su extensión escalable (SHVC). En algunos ejemplos, el codificador de vídeo 20 y decodificador de vídeo 30 operan de acuerdo con otros estándares, incluyendo estándares actualmente en desarrollo.

En HEVC y otras especificaciones de codificación de vídeo, los datos de vídeo incluyen una serie de imágenes. Las imágenes también pueden denominarse como "marcos". Una imagen puede incluir uno o más arreglos de muestras. Cada arreglo de muestras respectivo de una imagen puede comprender un arreglo de muestras para un componente de color respectivo. Una imagen puede incluir tres arreglos de muestras, denotados S<sub>L</sub>, S<sub>Cb</sub> y

$S_{Cr}$ ,  $S_L$  es un arreglo bidimensional (es decir, un bloque) de muestras de luma.  $S_{Cb}$  es un arreglo bidimensional de muestras de croma Cb.  $S_{Cr}$  es un arreglo bidimensional de muestras de croma Cr. En otros casos, una imagen puede ser monocromática y puede incluir solo un arreglo de muestras de luma.

- 5 Como parte de codificación de datos de vídeo, el codificador de vídeo 20 puede codificar imágenes de los datos de vídeo. En otras palabras, el codificador de vídeo 20 puede generar representaciones codificadas de las imágenes de los datos de vídeo. Una representación codificada de una imagen puede denominarse en este documento como una "imagen codificada" o "imagen cifrada".
- 10 Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede codificar bloques de la imagen. El codificador de vídeo 20 puede incluir, en un flujo de bits, una representación codificada del bloque de vídeo. En algunos ejemplos, para codificar un bloque de la imagen, el codificador de vídeo 20 realiza una intrapredicción o interpredicción para generar uno o más bloques predictivos. Adicionalmente, el codificador de vídeo 20 puede generar datos residuales para el bloque. El bloque residual comprende muestras residuales.
- 15 Cada muestra residual puede indicar una diferencia entre una muestra de uno de los bloques predictivos generados y una muestra correspondiente del bloque. El codificador de vídeo 20 puede aplicar una transformada a bloques de muestras residuales para generar coeficientes de transformada. Además, el codificador de vídeo 20 puede cuantificar los coeficientes de transformada. En algunos ejemplos, el codificador de vídeo 20 puede generar uno o más elementos de sintaxis para representar un coeficiente de transformada.
- 20 El codificador de vídeo 20 puede codificar por entropía uno o más de los elementos de sintaxis que representan el coeficiente de transformada.
- Más específicamente, al codificar datos de vídeo de acuerdo con HEVC u otras especificaciones de codificación de vídeo, para generar una representación codificada de una imagen, el codificador de vídeo 20 puede dividir
- 25 cada arreglo de muestra de la imagen en bloques de árbol de codificación (CTB) y codificar los CTB. Un CTB puede ser un bloque  $N \times N$  de muestras en un arreglo de muestra de una imagen. En el perfil principal de HEVC, el tamaño de un CTB puede oscilar desde  $16 \times 16$  a  $64 \times 64$ , aunque técnicamente se pueden soportar tamaños de CTB de  $8 \times 8$ .
- 30 Una unidad de árbol de codificación (CTU) de una imagen puede comprender uno o más CTB y puede comprender estructuras de sintaxis usadas para codificar las muestras del uno o más CTB. Por ejemplo, cada CTU puede comprender un CTB de muestras de luma, dos CTB correspondientes de muestras de croma y estructuras de sintaxis usadas para codificar las muestras de los CTB. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un único CTB y estructuras
- 35 de sintaxis usadas para codificar las muestras del CTB. Una CTU también puede denominarse como un "bloque de árbol" o una "unidad de codificación más grande" (LCU). En esta divulgación, una "estructura de sintaxis" puede definirse como cero o más elementos de sintaxis presentes juntos en un flujo de bits en un orden especificado. En algunos códecs, una imagen codificada es una representación codificada que contiene todas las CTU de la imagen.
- 40 Para codificar una CTU de una imagen, el codificador de vídeo 20 puede dividir los CTB de la CTU en uno o más bloques de codificación. Un bloque de codificación es un bloque  $N \times N$  de muestras. En algunos códecs, para codificar una CTU de una imagen, el codificador de vídeo 20 puede realizar de manera recursiva una división en árbol cuádruple en los bloques de árbol de codificación de una CTU para dividir los CTB en bloques
- 45 de codificación, por tanto el nombre de "unidades de árbol de codificación". Una unidad de codificación (CU) puede comprender uno o más bloques de codificación y estructuras de sintaxis usadas para codificar muestras del uno o más bloques de codificación. Por ejemplo, una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene un arreglo de muestras de luma, un arreglo de muestras Cb y un arreglo de muestras Cr, y estructuras
- 50 de sintaxis usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras de sintaxis usadas para codificar las muestras del bloque de codificación.
- Además, el codificador de vídeo 20 puede codificar CU de una imagen de los datos de vídeo. En algunos
- 55 códecs, como parte de la codificación de una CU, el codificador de vídeo 20 puede dividir un bloque de codificación de la CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras sobre el cual se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender uno o más bloques de predicción de una CU y estructuras de sintaxis usadas para predecir el uno o más bloques de predicción. Por ejemplo, una PU puede comprender un
- 60 bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras de sintaxis usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras de sintaxis usadas para predecir el bloque de predicción.
- 65 El codificador de vídeo 20 puede generar un bloque predictivo (por ejemplo, un bloque predictivo de luma, Cb y Cr) para un bloque de predicción (por ejemplo, un bloque de predicción de luma, Cb y Cr) de una PU de una CU. El codificador de vídeo 20 puede usar la intrapredicción o interpredicción para generar un bloque predictivo.

Si el codificador de vídeo 20 usa la intrapredicción para generar un bloque predictivo, el codificador de vídeo 20 puede generar el bloque predictivo con base en muestras decodificadas de la imagen que incluye la CU. Si el codificador de vídeo 20 usa la interpredicción para generar un bloque predictivo de una PU de una imagen actual, el codificador de vídeo 20 puede generar el bloque predictivo de la PU con base en muestras decodificadas de una imagen de referencia (es decir, una imagen distinta de la imagen actual). En HEVC, el codificador de vídeo 20 genera una estructura de sintaxis "prediction\_unit" dentro de una estructura de sintaxis "coding\_unit" para las PU interpredichas, pero no genera una estructura de sintaxis "prediction\_unit" dentro de una estructura de sintaxis "coding\_unit" para las PU intrapredichas. En cambio, en HEVC, los elementos de sintaxis relacionados con las PU intrapredichas se incluyen directamente en la estructura de sintaxis "coding\_unit".

El codificador de vídeo 20 puede generar uno o más bloques residuales para una CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de luma predictivos de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual Cb para la CU. Cada muestra en el bloque residual Cb de una CU puede indicar una diferencia entre una muestra Cb en uno de los bloques Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual Cr para la CU. Cada muestra en el bloque residual Cr de la CU puede indicar una diferencia entre una muestra Cr en uno de los bloques Cr predictivos de la CU y una muestra correspondiente en el bloque de codificación Cr original de la CU.

Además, el codificador de vídeo 20 puede descomponer los bloques residuales de una CU en uno o más bloques de transformada. Por ejemplo, el codificador de vídeo 20 puede usar la división de árbol cuádruple para descomponer los bloques residuales de una CU en uno o más bloques de transformada. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras sobre el cual se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender uno o más bloques de transformada. Por ejemplo, una TU puede comprender un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras de sintaxis usadas para transformar las muestras de bloque de transformada. De este modo, cada TU de una CU puede tener un bloque de transformada de luma, un bloque de transformada Cb y un bloque de transformada Cr. El bloque de transformada de luma de la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada Cb puede ser un subbloque del bloque residual Cb de la CU. El bloque de transformada Cr puede ser un subbloque del bloque residual Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras de sintaxis usadas para transformar las muestras del bloque de transformada.

El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de una TU para generar un bloque de coeficientes para la TU. Un bloque de coeficientes puede ser un arreglo bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. En algunos ejemplos, la una o más transformadas convierten el bloque de transformada de un dominio de píxeles a un dominio de frecuencia. De este modo, en tales ejemplos, un coeficiente de transformada puede ser una cantidad escalar que se considera que está en un dominio de frecuencia. Un nivel de coeficiente de transformada es una cantidad entera que representa un valor asociado con un índice de frecuencia bidimensional particular en un proceso de decodificación antes del escalamiento para el cálculo de un valor de coeficiente de transformada.

En algunos ejemplos, el codificador de vídeo 20 omite la aplicación de las transformadas al bloque de transformada. En tales ejemplos, el codificador de vídeo 20 puede tratar los valores de muestra residuales de la misma forma que los coeficientes de transformada. De este modo, en los ejemplos donde el codificador de vídeo 20 omite la aplicación de las transformadas, la siguiente discusión de los coeficientes de transformada y los bloques de coeficientes puede ser aplicable a los bloques de transformada de muestras residuales.

Después de generar un bloque de coeficientes, el codificador de vídeo 20 puede cuantificar el bloque de coeficientes para reducir posiblemente la cantidad de datos usados para representar el bloque de coeficientes, lo que potencialmente proporciona una compresión adicional. La cuantificación generalmente se refiere a un proceso en el cual un rango de valores se comprime a un único valor. Por ejemplo, la cuantificación se puede realizar dividiendo un valor por una constante y luego redondeando al entero más cercano. Para cuantificar el bloque de coeficientes, el codificador de vídeo 20 puede cuantificar los coeficientes de transformada del bloque de coeficientes. En algunos ejemplos, el codificador de vídeo 20 omite la cuantificación.

El codificador de vídeo 20 puede generar elementos de sintaxis que indican algunos o todos los coeficientes de transformada potencialmente cuantificados. El codificador de vídeo 20 puede codificar por entropía uno o más de los elementos de sintaxis que indican un coeficiente de transformada cuantificado. Por ejemplo, el codificador de vídeo 20 puede realizar una Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) sobre los elementos de sintaxis que indican los coeficientes de transformada cuantificados. De este modo, un



bloque codificado (por ejemplo, una CU codificada) puede incluir los elementos de sintaxis codificados por entropía que indican los coeficientes de transformada cuantificados.

El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos de vídeo codificados. En otras palabras, el codificador de vídeo 20 puede emitir un flujo de bits que incluye una representación codificada de datos de vídeo. La representación codificada de los datos de vídeo puede incluir una representación codificada de imágenes de los datos de vídeo. Por ejemplo, el flujo de bits puede comprender una secuencia de bits que forma una representación de imágenes codificadas de los datos de vídeo y datos asociados. En algunos ejemplos, una representación de una imagen codificada puede incluir representaciones codificadas de bloques de la imagen.

Un flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad de NAL es una estructura de sintaxis que contiene una indicación del tipo de datos en la unidad de NAL y bytes que contienen esos datos en la forma de una carga útil de secuencia de bytes sin procesar (RBSP) intercalada según sea necesario con bits de prevención de emulación. Cada una de las unidades de NAL puede incluir un encabezado de unidad de NAL y puede encapsular una RBSP. El encabezado de unidad de NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad de NAL. El código de tipo de unidad de NAL especificado por el encabezado de unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una estructura de sintaxis que contiene un número entero de bytes que está encapsulado dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Como se anotó anteriormente, el flujo de bits puede comprender una representación codificada de datos de vídeo. El decodificador de vídeo 30 puede decodificar el flujo de bits para reconstruir imágenes de los datos de vídeo. Como parte de la decodificación del flujo de bits, el decodificador de vídeo 30 puede obtener elementos de sintaxis del flujo de bits. El decodificador de vídeo 30 puede reconstruir imágenes de los datos de vídeo con base al menos en parte en los elementos de sintaxis obtenidos del flujo de bits. El proceso para reconstruir imágenes de los datos de vídeo puede ser generalmente recíproco al proceso realizado por el codificador de vídeo 20 para codificar las imágenes.

Por ejemplo, como parte de la decodificación de una imagen de los datos de vídeo, el decodificador de vídeo 30 puede usar la interpredicción o intrapredicción para generar bloques predictivos. Adicionalmente, el decodificador de vídeo 30 puede determinar los coeficientes de transformada con base en elementos de sintaxis obtenidos del flujo de bits. En algunos ejemplos, el decodificador de vídeo 30 cuantifica de manera inversa los coeficientes de transformada determinados. Además, el decodificador de vídeo 30 puede aplicar una transformada inversa en los coeficientes de transformada determinados para determinar los valores de las muestras residuales. El decodificador de vídeo 30 puede reconstruir un bloque de la imagen con base en las muestras residuales y las muestras correspondientes de los bloques predictivos generados. Por ejemplo, el decodificador de vídeo 30 puede agregar muestras residuales a las muestras correspondientes de los bloques predictivos generados para determinar las muestras reconstruidas del bloque.

Más específicamente, en HEVC y otras especificaciones de codificación de vídeo, el decodificador de vídeo 30 puede usar interpredicción o intrapredicción para generar uno o más bloques predictivos para cada PU de una CU actual. Además, el decodificador de vídeo 30 puede cuantificar de manera inversa los bloques de coeficientes de las TU de la CU actual. El decodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir bloques de transformada de las TU de la CU actual. El decodificador de vídeo 30 puede reconstruir un bloque de codificación de la CU actual con base en muestras de los bloques predictivos de las PU de la CU actual y muestras residuales de los bloques de transformada de las TU de la CU actual. En algunos ejemplos, el decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual agregando las muestras de los bloques predictivos para las PU de la CU actual a las muestras decodificadas correspondientes de los bloques de transformada de las TU de la CU actual. Al reconstruir los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

Como se mencionó anteriormente, un codificador de vídeo (por ejemplo, codificador de vídeo 20 o decodificador de vídeo 30) puede aplicar la interpredicción para generar un bloque predictivo para un bloque de vídeo de una imagen actual. Por ejemplo, el codificador de vídeo puede aplicar la interpredicción para generar un bloque de predicción de una CU. Si el codificador de vídeo aplica la interpredicción para generar un bloque de predicción, el codificador de vídeo genera el bloque de predicción con base en muestras decodificadas de una o más imágenes de referencia. Típicamente, las imágenes de referencia son imágenes distintas de la imagen actual. En algunas especificaciones de codificación de vídeo, un codificador de vídeo también puede tratar la propia imagen actual como una imagen de referencia.

Cuando un codificador de vídeo (por ejemplo, codificador de vídeo 20 o decodificador de vídeo 30) comienza a procesar una imagen actual, el codificador de vídeo puede determinar uno o más subconjuntos de conjuntos de imágenes de referencia (RPS) para la imagen actual. Por ejemplo, en HEVC, un codificador de vídeo puede determinar los siguientes subconjuntos de RPS: RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetFoll, RefPicSetLtCurr, y RefPicSetLtFoll. Además, el codificador de vídeo puede determinar una o más listas de

imágenes de referencia. Cada una de las listas de imágenes de referencia de una imagen actual incluye cero o más imágenes de referencia del RPS de la imagen actual. Una de las listas de imágenes de referencia puede denominarse como Lista de Imágenes de Referencia 0 (RefPicList0) y otra lista de imágenes de referencia puede denominarse como Lista de Imágenes de Referencia 1 (RefPicList1).

Una sección de una imagen puede incluir un número entero de bloques de la imagen. Por ejemplo, en HEVC y otras especificaciones de codificación de vídeo, una sección de una imagen puede incluir un número entero de CTU de la imagen. Las CTU de una sección pueden ordenarse consecutivamente en un orden de escaneo, tal como un orden de escaneo de barrido. En HEVC y otros estándares de codificación de vídeo, una sección se define como un número entero de CTU contenidas en un segmento de sección independiente y todos los segmentos de sección dependientes subsecuentes (si hay) que preceden al siguiente segmento de sección independiente (si hay) dentro de la misma unidad de acceso. Además, en HEVC y otros estándares de codificación de vídeo, un segmento de sección se define como un número entero de CTU ordenadas consecutivamente en el escaneo de mosaico y contenidas en una única unidad de NAL. Un escaneo de mosaico es un ordenamiento secuencial específico de CTB que dividen una imagen en la cual los CTB se ordenan consecutivamente en el escaneo de barrido de CTB en un mosaico, mientras que los mosaicos en una imagen se ordenan consecutivamente en un escaneo de barrido de los mosaicos de la imagen. Un mosaico es una región rectangular de CTB dentro de una columna de mosaicos particular y una fila de mosaicos particular en una imagen.

Como se anotó anteriormente, un flujo de bits puede incluir una representación de imágenes codificadas de los datos de vídeo y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros. Las unidades de NAL pueden encapsular RBSP para conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS) y conjuntos de parámetros de imagen (PPS). Un VPS es una estructura de sintaxis que comprende elementos de sintaxis que se aplican a cero o más secuencias de vídeo codificadas (CVS) completas. Un SPS también es una estructura de sintaxis que comprende elementos de sintaxis que se aplican a cero o más CVS completas. Un SPS puede incluir un elemento de sintaxis que identifica un VPS que está activo cuando el SPS está activo. De este modo, los elementos de sintaxis de un VPS pueden ser de aplicación más general que los elementos de sintaxis de un SPS. Un PPS es una estructura de sintaxis que comprende elementos de sintaxis que se aplican a cero o más imágenes codificadas. Un PPS puede incluir un elemento de sintaxis que identifica un SPS que está activo cuando el PPS está activo. Un encabezado de sección de un segmento de sección puede incluir un elemento de sintaxis que indica un PPS que está activo cuando está siendo codificado el segmento de sección.

Como se discutió anteriormente, un codificador de vídeo puede generar un flujo de bits que comprende una serie de unidades de NAL. En la codificación de vídeo multicapa, diferentes unidades de NAL del flujo de bits pueden estar asociadas con diferentes capas del flujo de bits. Una capa puede definirse como un conjunto de unidades de NAL de VCL y unidades de NAL de no VCL asociadas que tienen el mismo identificador de capa. Una capa puede ser equivalente a una vista en la codificación de vídeo multivista. En la codificación de vídeo multivista, una capa puede contener todos los componentes de vista de la misma capa con diferentes instancias de tiempo. Cada componente de vista puede ser una imagen codificada de la escena de vídeo que pertenece a una vista específica en una instancia de tiempo específica. En la codificación de vídeo multicapa, el término "unidad de acceso" puede referirse a un conjunto de imágenes que corresponden a la misma instancia de tiempo. De este modo, un "componente de vista" puede ser una representación codificada de una vista en una única unidad de acceso. En algunos ejemplos, un componente de vista puede comprender un componente de vista de textura (es decir, una imagen de textura) o un componente de vista de profundidad (es decir, una imagen de profundidad).

En algunos ejemplos de codificación de vídeo multivista, una capa puede contener ya sea todas las imágenes de profundidad codificadas de una vista específica o imágenes de textura codificadas de una vista específica. En otros ejemplos de codificación de vídeo multivista, una capa puede contener tanto componentes de vista de textura como componentes de vista de profundidad de una vista específica. De manera similar, en el contexto de la codificación de vídeo escalable, una capa corresponde típicamente a imágenes codificadas que tienen características de vídeo diferentes de las imágenes codificadas en otras capas. Tales características de vídeo típicamente incluyen resolución espacial y nivel de calidad (por ejemplo, Relación de Señal a Ruido).

Para cada capa respectiva del flujo de bits, los datos en una capa inferior pueden decodificarse sin referencia a los datos en cualquier capa superior. En la codificación de vídeo escalable, por ejemplo, los datos en una capa base pueden decodificarse sin referencia a los datos en una capa de mejora. En general, las unidades de NAL solo pueden encapsular datos de una única capa. De este modo, las unidades de NAL que encapsulan datos de la capa restante más alta del flujo de bits pueden retirarse del flujo de bits sin afectar la capacidad de decodificación de los datos en las capas restantes del flujo de bits. En la codificación multivista, las capas superiores pueden incluir componentes de vista adicionales. En SHVC, las capas superiores pueden incluir datos de mejora de relación de señal a ruido (SNR), datos de mejora espacial y/o datos de mejora temporal. En MV-HEVC y SHVC, una capa puede denominarse como una "capa base" si un decodificador de vídeo puede decodificar imágenes en la capa sin referencia a los datos de ninguna otra capa. La capa base puede cumplir con la especificación base de HEVC (por ejemplo, Rec. ITU-T H.265 | ISO/IEC 23008-2).

En la codificación de vídeo escalable, las capas distintas de la capa base pueden denominarse como "capas de mejora" y pueden proporcionar información que mejore la calidad visual de los datos de vídeo decodificados a partir del flujo de bits. La codificación de vídeo escalable puede mejorar la resolución espacial, relación de señal a ruido (es decir, calidad) o tasa temporal.

La codificación multivista puede soportar la predicción intervista. La predicción intervista es similar a la interpredicción usada en HEVC y puede usar los mismos elementos de sintaxis. Sin embargo, cuando un codificador de vídeo realiza una predicción intervista en una unidad de vídeo actual (tal como una PU), el codificador de vídeo 20 puede usar, como una imagen de referencia, una imagen que está en la misma unidad de acceso que la unidad de vídeo actual, pero en una vista diferente. Por el contrario, la interpredicción convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia.

En la codificación multivista, una vista puede denominarse como una "vista base" si un decodificador de vídeo (por ejemplo, decodificador de vídeo 30) puede decodificar imágenes en la vista sin referencia a imágenes en ninguna otra vista. Al codificar una imagen en una de las vistas no base, un codificador de vídeo (tal como codificador de vídeo 20 o decodificador de vídeo 30) puede agregar una imagen a una lista de imágenes de referencia si la imagen está en una vista diferente pero dentro de una misma instancia de tiempo (es decir, unidad de acceso) que la imagen que el codificador de vídeo está codificando actualmente. Al igual que otras imágenes de referencia de interpredicción, el codificador de vídeo puede insertar una imagen de referencia de predicción intervista en cualquier posición de una lista de imágenes de referencia.

Por ejemplo, las unidades de NAL pueden incluir encabezados (es decir, encabezados de unidad de NAL) y cargas útiles (por ejemplo, RBSP). Los encabezados de unidad de NAL pueden incluir elementos de sintaxis de identificador de capa (por ejemplo, elementos de sintaxis `nuh_layer_id` en HEVC). Las unidades de NAL que tienen elementos de sintaxis de identificador de capa que especifican diferentes valores pertenecen a diferentes "capas" de un flujo de bits. De este modo, en la codificación de vista multicapa (por ejemplo, MV-HEVC, SVC o SHVC), el elemento de sintaxis de identificador de capa de la unidad de NAL especifica un identificador de capa (es decir, un ID de capa) de la unidad de NAL. El identificador de capa de una unidad de NAL es igual a 0 si la unidad de NAL se relaciona con una capa base en la codificación multicapa. Los datos en una capa base de un flujo de bits se pueden decodificar sin referencia a los datos en cualquier otra capa del flujo de bits. Si la unidad de NAL no se relaciona con una capa base en la codificación multicapa, el identificador de capa de la unidad de NAL puede tener un valor de no cero. En la codificación multivista, las diferentes capas de un flujo de bits pueden corresponder a diferentes vistas. En la codificación de vídeo escalable (por ejemplo, SVC o SHVC), las capas distintas de la capa base pueden denominarse como "capas de mejora" y pueden proporcionar información que mejore la calidad visual de los datos de vídeo decodificados a partir del flujo de bits.

Además, algunas imágenes dentro de una capa pueden decodificarse sin referencia a otras imágenes dentro de la misma capa. De este modo, las unidades de NAL que encapsulan datos de ciertas imágenes de una capa pueden retirarse del flujo de bits sin afectar la capacidad de decodificación de otras imágenes en la capa. El retiro de unidades de NAL que encapsulan datos de tales imágenes puede reducir la tasa de marcos del flujo de bits. Un subconjunto de imágenes dentro de una capa que puede decodificarse sin referencia a otras imágenes dentro de la capa puede denominarse en este documento como una "subcapa", "capa temporal" o una "subcapa temporal". Una capa temporal más alta puede incluir todas las imágenes en la capa. De este modo, la escalabilidad temporal puede lograrse dentro de una capa definiendo un grupo de imágenes con un nivel temporal particular como una subcapa (es decir, capa temporal).

Las unidades de NAL pueden incluir elementos de sintaxis de identificador temporal (por ejemplo, `temporal_id` en HEVC). El elemento de sintaxis de identificador temporal de una unidad de NAL especifica un identificador temporal de la unidad de NAL. El identificador temporal de una unidad de NAL identifica una subcapa temporal con la cual está asociada la unidad de NAL. De este modo, cada subcapa temporal de un flujo de bits puede estar asociada con un identificador temporal diferente. Si el identificador temporal de una primera unidad de NAL es menor que el identificador temporal de una segunda unidad de NAL, los datos encapsulados por la primera unidad de NAL pueden decodificarse sin referencia a los datos encapsulados por la segunda unidad de NAL.

Los estándares de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones Codificación de Vídeo Escalable (SVC) y Codificación de Vídeo Multivista (MVC). Además, recientemente se ha desarrollado un nuevo estándar de codificación de vídeo, a saber Codificación de Vídeo de Alta Eficiencia (HEVC), por el Equipo de Colaboración Conjunta sobre 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. Wang et al., "High Efficiency Video Coding (HEVC) Defect Report", Equipo Colaborativo Conjunto sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 14<sup>o</sup> Reunión, Viena, AT, 25 de Julio - 2 de agosto 2013, documento JCTVC-N1003-v1, es una especificación de borrador de HEVC. El estándar de HEVC se ha finalizado en enero de 2013.

ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG 11) están estudiando actualmente la potencial necesidad de estandarización de la futura tecnología de codificación de vídeo con una capacidad de compresión que exceda significativamente la del estándar de HEVC actual (incluyendo sus extensiones actuales y las extensiones a corto plazo para la codificación de contenido de pantalla y la codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo de colaboración conjunta conocido como el Equipo de Exploración Conjunta de Vídeo (JVET) para evaluar los diseños de tecnología de compresión propuestos por sus expertos en esta área. El JVET se reunió por primera vez entre el 19-21 de octubre de 2015. El Modelo Conjunto de Exploración (JEM) es un modelo de prueba producido por el JVET. J. Chen et al., "Description of Exploration Experiments on Coding Tools", JVET-D1011, Chengdu, octubre de 2016 es una descripción de algoritmo para la cuarta versión de JEM (es decir, JEM4).

En el campo de la codificación de vídeo, es habitual aplicar un filtrado con el fin de mejorar la calidad de una señal de vídeo decodificada. El filtro se puede aplicar como un posfiltro, donde el marco filtrado no se usa para la predicción de marcos futuros o como un filtro en bucle, donde el marco filtrado se usa para predecir marcos futuros. Un filtro se puede diseñar, por ejemplo, minimizando el error entre la señal original y la señal filtrada decodificada. Similar a los coeficientes de transformada, los coeficientes del filtro  $h(k, l)$ ,  $k = -K, \dots, K$ ,  $l = -K, \dots, K$  se pueden cuantificar de la siguiente manera:

$$f(k, l) = \text{redondo}(\text{normFactor} \cdot h(k, l))$$

y se codifican y se envían a un decodificador. El normFactor usualmente es igual a  $2^n$ . Cuanto mayor sea el valor de normFactor, más precisa será la cuantificación y los coeficientes de filtro cuantificados  $f(k, l)$  proporcionan un mejor rendimiento. Por otro lado, los valores mayores de normFactor producen coeficientes  $f(k, l)$  que requieren más bits para transmitir.

En el decodificador de vídeo 30, los coeficientes de filtro decodificados  $f(k, l)$  se aplican a la imagen reconstruida  $R(i, j)$  de la siguiente manera:

$$\tilde{R}(i, j) = \sum_{k=-K}^K \sum_{l=-K}^K f(k, l) R(i + k, j + l) / \sum_{k=-K}^K \sum_{l=-K}^K f(k, l), \quad (1)$$

donde  $i$  y  $j$  son las coordenadas de los píxeles dentro del marco. El filtro adaptativo en bucle fue evaluado en la etapa de HEVC, pero no se incluyó en la versión final.

El filtro de bucle adaptativo en bucle empleado en el JEM fue descrito en J. Chen et al., "Coding tools investigation for next generation video coding", SG16-Ginebra-C806, enero de 2015. La idea básica es la misma que la del ALF con adaptación basada en bloques en T. Wiegand et al., "WD3: Working Draft 3 of High-Efficiency Video Coding", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, JCTVC-E603, 5ª Reunión: Ginebra, Suiza, 16-23 de marzo de 2011, de aquí en adelante, "JCTVC-E603".

Para el componente de luma, los bloques 4x4 en la imagen completa se clasifican con base en la dirección Laplaciana unidimensional (hasta 3 direcciones) y la actividad Laplaciana bidimensional (hasta 5 valores de actividad). El cálculo de la dirección  $Dir_b$  y la actividad no cuantificada  $Act_b$  se muestra en las ecuaciones (2) hasta (5), donde  $\hat{I}_{i,j}$  indica un píxel reconstruido con la coordenada relativa  $(i, j)$  en la parte superior izquierda de un bloque 4x4.  $Act_b$  se cuantifica además en el rango de 0 a 4, de manera inclusive, como se describe en JCTVC-E603.

$$V_{i,j} = |\hat{I}_{i,j} \times 2 - \hat{I}_{i,j-1} - \hat{I}_{i,j+1}| \quad (2)$$

$$H_{i,j} = |\hat{I}_{i,j} \times 2 - \hat{I}_{i-1,j} - \hat{I}_{i+1,j}| \quad (3)$$

$$Dir_b = \begin{cases} 1, & \text{si } (\sum_{i=0}^3 \sum_{j=0}^3 H_{i,j} > 2 \times \sum_{i=0}^3 \sum_{j=0}^3 V_{i,j}) \\ 2, & \text{si } (\sum_{i=0}^3 \sum_{j=0}^3 V_{i,j} > 2 \times \sum_{i=0}^3 \sum_{j=0}^3 H_{i,j}) \\ 0, & \text{de otro modo} \end{cases} \quad (4)$$

$$Act_b = \sum_{i=0}^3 \sum_{j=0}^3 (\sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} (V_{m,n} + H_{m,n})) \quad (5)$$

En total, cada bloque puede categorizarse en uno de 15 grupos (5x3) y se asigna un índice a cada bloque 4x4 de acuerdo con el valor de  $Dir_b$  y  $Act_b$  del bloque. Denotar el índice de grupo por  $C$  y establecer  $C$  igual a  $5Dir_b + \hat{A}$  en donde  $\hat{A}$  es el valor cuantificado de  $Act_b$ . Por lo tanto, el codificador de vídeo 20 puede señalar hasta 15 conjuntos de parámetros de ALF para el componente de luma de una imagen. Para ahorrar el coste de señalización, el codificador de vídeo 20 puede fusionar los grupos a lo largo del valor de índice de grupo. Para cada grupo fusionado, el codificador de vídeo 20 puede señalar un conjunto de coeficientes de ALF. La figura 2 ilustra tres diferentes soportes de filtros de ALF de ejemplo. En el ejemplo de la figura 2, se soportan hasta tres conformaciones de filtro simétricas circulares. Para ambos componentes de croma en una imagen, se aplica un único conjunto de coeficientes de ALF y siempre se usa el filtro de conformación de diamante 5x5.

En el lado de decodificador, el decodificador de vídeo 30 puede filtrar cada muestra de píxel  $\hat{I}_{i,j}$ , dando como resultado un valor de píxel  $I'_{i,j}$  como se muestra en la ecuación (6), donde  $L$  denota la longitud de filtro,  $f_{m,n}$  representa el coeficiente de filtro y  $o$  indica la compensación de filtro.

$$I'_{i,j} = \sum_{m=-L}^L \sum_{n=-L}^L f_{m,n} \times \hat{I}_{i+m,j+n} + o \quad (6)$$

En algunos diseños, solo se soporta hasta un filtro para los dos componentes de croma.

La siguiente es una lista de datos que pueden señalarse para los coeficientes de filtro.

1. Número total de filtros: Número total de filtros (o número total de grupos fusionados) se señala en primer lugar cuando se habilita ALF para una sección. El número total de filtros señalado se aplica al componente de luma. Para los componentes de croma, dado que solo se puede aplicar un filtro de ALF, no hay necesidad de señalar el número total de filtros.

2. Soporte de filtros: Se señala un índice de los tres soportes de filtros.

3. Índice de filtro: Indica qué filtro de ALF se usa, es decir, información de fusión de clases. Las clases que tienen valores no consecutivos de  $C$  se pueden fusionar, es decir, comparten el mismo filtro. Al codificar un indicador de cada clase para indicar si la clase está fusionada o no, se puede derivar el índice de filtro. En algunos ejemplos, la información de fusión de clases también se puede señalar para fusionarse desde un índice de filtro izquierdo o superior.

4. Indicador forceCoeff0: El indicador forceCoeff0 se usa para indicar si al menos uno de los filtros no debe codificarse. Cuando este indicador es igual a 0, se deben codificar todos los filtros. Cuando el indicador forceCoeff0 es igual a 1, se señala además un indicador de cada grupo fusionado, denotado por CodedVarBin, para indicar si se debe señalar o no el filtro. Cuando no se señala el filtro, significa que todos los coeficientes de filtro asociados con el filtro son iguales a 0.

5. Método de predicción: Cuando es necesario señalar múltiples grupos de filtros, se puede usar uno de los dos métodos:

- Todos los filtros se codifican directamente en la información de filtro. En este caso, por ejemplo, los valores de los coeficientes de filtro se pueden codificar en el flujo de bits sin usar ninguna técnica de codificación predictiva. En otras palabras, los filtros se señalizan explícitamente.

- Los coeficientes de filtro de un primer filtro se codifican directamente. Mientras que para los filtros restantes, los coeficientes de filtro se codifican de manera predictiva en la información de filtro. En este caso, los valores de los coeficientes de filtro pueden definirse mediante valores residuales o diferencias relativas a los coeficientes de filtro asociados con un filtro codificado previamente. El filtro codificado previamente es el que es el filtro más reciente (es decir, los índices de filtro del filtro actual y su predictor son consecutivos).

Para indicar el uso de uno de los dos métodos de predicción anteriores, el codificador de vídeo 20 puede señalar un indicador cuando el número de grupos fusionados es mayor que 1 y forceCoeff0 es igual a 0.

Un conjunto de unos parámetros de ALF puede incluir uno o más de los elementos de sintaxis enumerados anteriormente y también puede incluir coeficientes de filtro.

Un codificador de vídeo (por ejemplo, codificador de vídeo 20 o decodificador de vídeo 30) también puede usar la predicción temporal de los coeficientes de filtro. El codificador de vídeo puede almacenar los coeficientes de ALF de imágenes codificadas previamente y puede reutilizar los coeficientes de ALF de las imágenes codificadas previamente como coeficientes de ALF de una imagen actual. El codificador de vídeo 20 puede elegir usar los coeficientes de ALF almacenados para la imagen actual y evitar la señalización de los coeficientes de ALF. En este caso, el codificador de vídeo 20 solo señala un índice a una de las imágenes de referencia (que en realidad es igual al índice del candidato en el arreglo almacenado para los parámetros de

ALF), y los coeficientes de ALF almacenados de la imagen indicada simplemente se heredan para la imagen actual. Para indicar el uso de la predicción temporal, el codificador de vídeo 20 puede codificar primero un indicador que indique el uso de la predicción temporal, antes de enviar el índice a la imagen de referencia.

5 En JEM4, los codificadores de vídeo almacenan los parámetros de ALF de seis imágenes codificadas previamente como máximo que se codifican con parámetros de ALF señalizados (es decir, predicción temporal está deshabilitada) en un arreglo separado. Un codificador de vídeo vacía efectivamente el arreglo para imágenes de intrapunto de acceso aleatorio (IRAP). Para evitar duplicados, el codificador de vídeo solo almacena valores de parámetros de ALF en el arreglo si los valores de parámetros de ALF fueron señalizados  
10 explícitamente. El almacenamiento de parámetros de ALF opera en un modo de FIFO, por lo que si el arreglo está lleno, el codificador de vídeo sobrescribe los valores de parámetros de ALF más antiguos (es decir, parámetros de ALF) con un nuevo conjunto de valores de parámetros de ALF, en orden de decodificación.

15 En M. Karczewicz et al., "Improvements on adaptive loop filter", Equipo de Exploración (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, Doc. JVET-B0060\_r1, 2da Reunión: San Diego, Estados Unidos, 20-26 de febrero de 2016 (de aquí en adelante, "JVET-B0060"), se propone el ALF basado en transformaciones Geométricas (GALF). En GALF, la clasificación se modifica con los gradientes diagonales tenidos en cuenta y se pueden aplicar transformaciones geométricas a los coeficientes de filtro.

20 Con base en toda la información de gradiente incluyendo los gradientes horizontales, verticales y diagonales, se determina una de las cuatro transformaciones geométricas de los coeficientes de filtro. Es decir, las muestras clasificadas en la misma categoría compartirán los mismos coeficientes de filtro. Sin embargo, la región de soporte de filtro puede transformarse con base en el índice de transformación geométrica seleccionado. El método descrito en JVET-B0060 puede reducir de manera efectiva el número de filtros que tienen que enviarse  
25 al decodificador, reduciendo por tanto el número de bits requeridos para representarlos, o alternativamente reducir las diferencias entre los marcos reconstruidos y los marcos originales. Cada bloque 2x2 se clasifica en una de las 25 clases con base en su direccionalidad y un valor cuantificado de actividad.

30 Además, en JVET-B0060, para mejorar la eficiencia de codificación cuando no está disponible la predicción temporal (por ejemplo, en intramarcos), un codificador de vídeo asigna un conjunto de 16 filtros fijos a cada clase. Es decir, se pueden predefinir 16\*25(clases) filtros. Para indicar el uso de un filtro fijo, se señala un indicador para cada clase y, si es requerido, el índice del filtro fijo. Incluso cuando se selecciona el filtro fijo para una clase dada, los coeficientes del filtro adaptativo  $f(k, l)$  todavía se pueden enviar para esta clase, en cuyo caso los coeficientes del filtro que se aplicarán a la imagen reconstruida son la suma de ambos conjuntos de  
35 coeficientes. Una o más de las clases pueden compartir los mismos coeficientes  $f(k, l)$  señalados en el flujo de bits, incluso si fueron elegidos diferentes filtros fijos para ellas. La Publicación de Patente de los Estados Unidos No. 2017/0238020, publicada el 17 de agosto de 2017, describe cómo los filtros fijos también podrían aplicarse a marcos intercodificados.

40 En JVET-B0060, el diseño de predicción temporal a partir de marcos codificados previamente como en la segunda versión de JEM (es decir, JEM2) se mantiene sin cambios. JEM2 se describe en Jianle Chen et al., "Algorithm Description of Joint Exploration Test Model 2", Equipo de Exploración Conjunta de Vídeo (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 2da Reunión, San Diego, USA, 20-26 de febrero de 2016, documento JVET-B1001\_v3. Es decir, se codifica un indicador para indicar si se usa la predicción temporal de los coeficientes de ALF. Si se usa la predicción temporal de los coeficientes de ALF, se señala además un  
45 índice de las imágenes seleccionadas para los parámetros de ALF almacenados. En este caso, no es necesario señalar los índices de filtro para cada clase y coeficientes de filtro.

50 Además, se puede usar la codificación explícita de los coeficientes de filtro de ALF con GALF. Por ejemplo, se puede codificar explícitamente un patrón de predicción y un índice de predicción de filtros fijos en GALF. Se definen tres casos:

- caso 1: si ninguno de los filtros de las 25 clases se predice a partir de los filtros fijos;

55 • caso 2: todos los filtros de las clases se predicen a partir de los filtros fijos; y

- caso 3: los filtros asociados con algunas clases se predicen a partir de los filtros fijos y los filtros asociados con el resto de las clases no se predicen a partir de los filtros fijos.

60 En primer lugar se puede codificar un índice para indicar uno de los tres casos. Además, se aplica lo siguiente:

- Si el caso indicado es el caso 1, no hay necesidad de señalar más el índice del filtro fijo.

- De otro modo, si el caso indicado es el caso 2, se señala un índice del filtro fijo seleccionado para cada clase.

65 • De otro modo, si el caso indicado es el caso 3, primero se señala un bit para cada clase, y si se usa un filtro fijo, se señala además el índice del filtro fijo.

En GALF, para reducir el número de bits requeridos para representar los coeficientes de filtro, se pueden fusionar diferentes clases. Sin embargo, a diferencia de JCTVC-E603, se puede fusionar cualquier conjunto de clases, incluso las clases que tienen valores de C no consecutivos. La información sobre qué clases se fusionan se proporciona enviando para cada una de las 25 clases un índice  $i_c$ . Las clases que tienen el mismo índice  $i_c$  comparten los mismos coeficientes de filtro que se codifican. El índice  $i_c$  se codifica con un método de longitud fija truncado.

De manera similar, también se puede usar el indicador forceCoeff0. Cuando el indicador forceCoeff0 es igual a 1, se señala además un indicador de un bit, denotado por CodedVarBin, para cada uno de los grupos fusionados (todos los filtros que van a ser codificados) para indicar si los coeficientes de filtro señalados son todos cero. Además, cuando forceCoeff0 es igual a 1, se desactiva la codificación predictiva (es decir, la codificación de la diferencia entre el filtro actual y el filtro codificado previamente). Cuando se permite la predicción a partir de filtros fijos, los filtros que van a ser señalados/codificados mencionados anteriormente son las diferencias entre el filtro aplicado a la imagen reconstruida y el filtro fijo seleccionado. Otra información, tales como los coeficientes se codifican de la misma forma como en JEM2.0.

Debido a que GALF es una forma de ALF, esta divulgación puede usar el término "ALF" para aplicarse tanto a ALF como a GALF.

Los diseños actuales para la predicción temporal de filtros en ALF y GALF tienen varias deficiencias. Por ejemplo, si una imagen usa una codificación explícita de filtros, después de decodificar la imagen, los filtros de ALF correspondientes pueden agregarse a un arreglo de filtros de ALF para la predicción temporal, independientemente de las capas temporales. Es decir, después de decodificar la imagen, un codificador de vídeo puede incluir un conjunto de parámetros de ALF en una entrada en el arreglo. El conjunto de parámetros de ALF puede incluir coeficientes de filtro e información de fusión de grupos para cada uno de los filtros de ALF usados en la imagen. Este diseño lleva a falla cuando se decodifica un subconjunto de capas temporales en determinadas configuraciones, tal como acceso aleatorio. En la figura 3 se da un ejemplo en donde el tamaño de GOP es igual a 16. En el ejemplo de la figura 3, se soportan cinco capas temporales (indicadas por  $T_0$  a  $T_4$ ). El orden de codificación/decodificación de las imágenes es: Conteo de Orden de Imagen (POC) 0 [ $T_0$ ], POC 16 [ $T_0$ ], POC8 [ $T_1$ ], POC4 [ $T_2$ ], POC2 [ $T_3$ ], POC1 [ $T_4$ ], POC3 [ $T_4$ ], POC6 [ $T_3$ ], POC5 [ $T_4$ ], POC7 [ $T_4$ ], POC12 [ $T_2$ ], POC10 [ $T_3$ ], POC9 [ $T_4$ ], POC11 [ $T_4$ ], POC14 [ $T_3$ ], POC13 [ $T_4$ ], POC15 [ $T_4$ ]. Las flechas con diferentes patrones de trazos señalan imágenes que pueden usar las imágenes señaladas como imágenes de referencia. Nótese que la figura 3 omite ciertas flechas en aras de la claridad.

La figura 4A ilustra un arreglo 50 para almacenar parámetros de filtro. La figura 4B ilustra un estado diferente del arreglo 50. Suponiendo que cada imagen está codificada con ALF habilitado y que los filtros de ALF para cada imagen están explícitamente señalizados, antes de decodificar POC3 de la figura 3, el arreglo para filtros almacenados tiene el estado que se muestra en la figura 4A. Después de decodificar POC3 y antes de decodificar POC6 de la figura 3, el arreglo para filtros de ALF almacenados se actualiza como se muestra en la figura 4B. Como se muestra en el ejemplo de la figura 4B, los filtros para POCO se han reemplazado con los filtros para POC3 debido a que los filtros se reemplazan en una forma de FIFO y los filtros para POCO fueron los primeros filtros agregados al arreglo 50.

Por lo tanto, para decodificar POC6 con un índice de capa temporal (Templdx) igual a 3, se requiere que se decodifiquen los filtros de POC1, POC3 con un índice de capa temporal igual a 4. Esto entra en conflicto con el espíritu de escalabilidad temporal, en donde la decodificación de una imagen con un cierto valor de Templdx no debería depender de imágenes con un valor mayor de Templdx.

Una segunda deficiencia de los diseños actuales para la predicción temporal de filtros en ALF es que, cuando se habilita la predicción temporal de filtros de ALF para una sección, en algunos ejemplos, se heredarán todos los filtros de ALF de un cierto marco codificado previamente. Esto significa que la fusión de clases y coeficientes de filtro se reutilizan directamente sin la posibilidad de modificar ligeramente las clases y los coeficientes de filtro para capturar mejor las características de una sección actual.

Se proponen las siguientes técnicas para resolver una o más de las deficiencias de los diseños actuales para la predicción temporal de filtros en ALF mencionadas anteriormente. Las siguientes técnicas detalladas pueden aplicarse individualmente. Alternativamente, puede aplicarse cualquier combinación de ellas.

De acuerdo con una primera técnica, se pueden asignar múltiples arreglos para almacenar uno o más conjuntos de filtros de ALF codificados previamente. En otras palabras, un codificador de vídeo puede almacenar conjuntos de parámetros de ALF en una pluralidad de arreglos. Cada arreglo corresponde a un índice de capa temporal asignado (Templdx, que es equivalente a TemporalId definido en la especificación de HEVC). De acuerdo con la primera técnica, cada arreglo solo contiene parámetros de ALF de imágenes con el mismo Templdx o Templdx inferior. Una sección (u otra unidad para realizar ALF) con el Templdx puede seleccionar un conjunto de filtros compuestos en este arreglo. En otras palabras, un codificador de vídeo puede aplicar, a muestras en un bloque de la sección, un filtro de ALF basado en parámetros de ALF en el arreglo que

corresponde al TempIdx de la sección. Para una región que está codificada con ALF habilitado, y suponiendo que los parámetros de ALF están explícitamente señalizados (es decir, sin predicción temporal), el conjunto de parámetros de ALF para esta región se puede agregar al arreglo asociado con el mismo o superior TempIdx. Esto puede resolver las deficiencias descritas anteriormente con respecto al arreglo de parámetros de ALF almacenados que incluyen uno o más parámetros de ALF que corresponden a filtros de ALF usados en imágenes de capas temporales superiores a la capa temporal de la imagen actual.

La figura 5 ilustra una pluralidad de arreglos 60A-60E (colectivamente, "arreglos 60") que corresponden a diferentes capas temporales, de acuerdo con una técnica de esta divulgación. En el ejemplo de la figura 5, suponiendo que cada imagen de la figura 3 está codificada con ALF habilitado y los filtros de ALF para cada imagen están señalizados explícitamente, antes de decodificar POC6 de la figura 3, los arreglos para los filtros de ALF almacenados tienen los estados que se muestran en la figura 5.

En el ejemplo de la figura 5, dado que POC6 está en la capa temporal  $T_3$ , un codificador de vídeo puede usar filtros de ALF del arreglo 60D. De este modo, a diferencia del ejemplo de la figura 4B, si se decodifica o no POC1 no tiene ningún impacto en los filtros de ALF que están disponibles para uso al decodificar POC 6.

De esta forma, de acuerdo con la primera técnica de la invención reivindicada, el codificador de vídeo 20 genera un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo. Una región actual (por ejemplo, sección u otro tipo de unidad para realizar ALF) de la imagen actual está asociada con un índice temporal (es decir, un índice de capa temporal) que indica una capa temporal a la cual pertenece la región actual. Además, el codificador de vídeo 20 reconstruye la totalidad o parte de la imagen actual. El codificador de vídeo 20 almacena, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo decodificados antes de la región actual. Para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, el codificador de vídeo 20 almacena, en el arreglo respectivo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo. Cada arreglo respectivo de la pluralidad de arreglos corresponde a una respectiva capa temporal diferente. Además, el codificador de vídeo 20 determina, con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En algunos ejemplos, el codificador de vídeo 20 puede determinar el conjunto seleccionado de parámetros de ALF con base en un análisis de distorsión de tasa de los conjuntos de parámetros de ALF en los arreglos. El codificador de vídeo 20 puede señalar un índice del conjunto seleccionado de parámetros de ALF en el flujo de bits. Además, el codificador de vídeo 20 aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual. La aplicación del filtrado de bucle adaptativo a la región actual puede comprender aplicar un filtro de ALF a uno o más bloques, pero no necesariamente a todos, dentro de la región actual. Después de aplicar el filtrado de bucle adaptativo a la región actual, el codificador de vídeo 20 puede usar la región actual para la predicción de una imagen subsecuente de los datos de vídeo.

De manera similar, de acuerdo con la primera técnica de la invención reivindicada, el decodificador de vídeo 30 recibe un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo. Una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual. El decodificador de vídeo 30 luego reconstruye la totalidad o parte de la imagen actual. Adicionalmente, el decodificador de vídeo 30 almacena, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo decodificados antes de la imagen actual. Cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, el decodificador de vídeo 30 almacena, en el arreglo respectivo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo. El decodificador de vídeo 30 determina, con base en un conjunto seleccionado de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual o arreglos de la pluralidad de arreglos que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En algunos ejemplos, el decodificador de vídeo 30 determina el conjunto seleccionado de parámetros de ALF con base en un índice señalado en el flujo de bits. El decodificador de vídeo 30 luego aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual. La aplicación del filtro de ALF a la región actual puede comprender la aplicación del filtro de ALF a uno o más bloques, pero no necesariamente a todos, dentro de la región actual.

Cada arreglo asignado por TempIdx puede comprender conjuntos de filtros decodificados previamente asociados con imágenes con TempIdx igual o inferior. Por ejemplo, el k-ésimo arreglo se asigna para asociarse



con TempIdx igual a k, y solo contendrá los conjuntos completos o los subconjuntos de filtros (por ejemplo, parámetros de ALF para filtros) de imágenes con TempIdx igual a o menor que k.

De este modo, para cada arreglo respectivo de la pluralidad de arreglos, un codificador de vídeo (por ejemplo, codificador de vídeo 20 o decodificador de vídeo 30) puede almacenar en el arreglo respectivo conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de las imágenes de los datos de vídeo decodificados antes de la imagen actual que pertenecen a la capa temporal que corresponde al arreglo respectivo y que pertenecen a capas temporales inferiores a la capa temporal que corresponde al arreglo respectivo.

En algunos ejemplos, los números de conjuntos de filtros asociados con diferentes arreglos pueden ser diferentes (que podrían ser mayores o iguales a 0). Alternativamente, en algunos ejemplos, los números de los conjuntos de filtros asociados con diferentes capas temporales pueden ser diferentes y pueden depender del índice de capa temporal. De este modo, en algunos ejemplos, al menos dos de la pluralidad de arreglos incluyen diferentes números de conjuntos de parámetros de ALF. Por ejemplo, en el ejemplo de la figura 5, puede ser innecesario tener cinco ubicaciones en el arreglo 60A debido a que, en un GOP de 16 imágenes, nunca habrá más de dos imágenes en la capa temporal  $T_0$ . De este modo, el arreglo 60A solo puede tener dos ubicaciones. De manera similar, en el ejemplo de la figura 5, en un GOP de 16 imágenes, habrá como máximo una imagen en la capa temporal  $T_1$ . Por tanto, el arreglo 60B solo puede tener tres ubicaciones.

En algunos ejemplos, después de codificar una cierta sección/unidad para realizar ALF, un codificador de vídeo puede usar el conjunto de filtros asociados con la sección para actualizar aquellos arreglos asociados con un TempIdx igual o superior. Por ejemplo, un codificador de vídeo puede almacenar, en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual (y, en algunos casos, arreglos que corresponden a capas temporales que corresponden a capas temporales superiores a la capa temporal a la cual pertenece la región actual), un conjunto de parámetros de ALF aplicables a una región actual (es decir, sección u otra unidad para realizar ALF). Por ejemplo, en el ejemplo de la figura 3 y figura 5, si la región actual está en una imagen asociada con POC 8, el codificador de vídeo puede actualizar los arreglos 60B, 60C, 60D y 60E para incluir el conjunto de parámetros de ALF aplicables a la región actual.

En algunos ejemplos, el valor de POC asociado con cada conjunto de filtros (por ejemplo, un conjunto de parámetros de ALF) también puede registrarse. De este modo, un codificador de vídeo puede almacenar, en el arreglo que corresponde a la capa temporal a la cual pertenece una región actual de una imagen actual, un valor de POC de la imagen actual. En un ejemplo, al seleccionar un filtro como un candidato de un arreglo dado para la predicción temporal de ALF, puede requerirse que el valor de POC asociado con el filtro sea igual a un valor de POC de una de las imágenes de referencia en las listas de imágenes de referencia actuales. Por ejemplo, además de almacenar los parámetros de ALF para los filtros de ALF usados por la imagen en POCO en la figura 5, un codificador de vídeo puede almacenar en el arreglo 60A datos que indican un valor de POCO. En este ejemplo, si la imagen en POCO no está en una imagen de referencia de la imagen en POC6 al codificar una región de la imagen en POC6, el codificador de vídeo 20 no tiene permitido seleccionar un filtro de ALF de entre los filtros de ALF almacenados en el arreglo 60A para la imagen en POCO.

De acuerdo con una segunda técnica, todavía se usa un arreglo para almacenar conjuntos de filtros de ALF codificados previamente. Además de los filtros, para cada conjunto (que puede contener múltiples filtros usados para codificar una sección/imagen), también se registra el índice de capa temporal (TempIdx) asociado con el conjunto de filtros. En otras palabras, los índices de capa temporal se pueden almacenar junto con los parámetros de ALF para los filtros de ALF.

En algunos ejemplos basados en la segunda técnica, el tamaño del arreglo puede establecerse en (número de capas temporales posibles)\*(número máximo de conjuntos de filtros para predicción temporal de una sección/imagen u otra unidad para uso de ALF). En un ejemplo, el número de capas temporales posibles puede depender de una estructura de codificación (por ejemplo, cuántos niveles se soportan en la estructura de jerarquía B) o un indicador de verificación de retraso bajo (NoBackwardPredFlag en la especificación de HEVC).

En un ejemplo, se puede predefinir o señalar un número máximo de conjuntos de filtros para la predicción temporal de una sección/imagen u otra unidad para uso de ALF o depender del TempIdx. En un ejemplo, el número de capas temporales posibles se establece en 5 y el número máximo de conjuntos de filtros para la predicción temporal de una sección/imagen u otra unidad para uso de ALF se establece en 6. Al codificar una sección/imagen, los posibles candidatos para la predicción temporal se pueden decidir recorriendo los conjuntos incluidos en el arreglo y todos o algunos conjuntos de filtros con TempIdx igual o menor se tratan como candidatos efectivos.

Después de codificar una cierta sección/unidad para realizar ALF, el conjunto de filtros asociados con la sección y el TempIdx asociado se pueden usar para actualizar el arreglo. Por ejemplo, un codificador de vídeo (por ejemplo, codificador de vídeo 20 o decodificador de vídeo 30) puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo, un conjunto aplicable de parámetros de ALF para una región. En este ejemplo, el codificador o decodificador puede almacenar, en el arreglo, el conjunto aplicable de

parámetros de ALF. El codificador o decodificador también puede almacenar el conjunto aplicable de parámetros de ALF en uno o más de los arreglos que corresponden a capas temporales superiores a la capa temporal a la cual pertenece la región actual. En este ejemplo, el codificador de vídeo no puede almacenar parámetros de ALF en el arreglo si los parámetros de ALF no fueron señalados explícitamente en el flujo de bits. En algunos ejemplos, el codificador o decodificador solo almacena el conjunto aplicable de parámetros de ALF en el arreglo si el conjunto aplicable de parámetros de ALF aún no se ha almacenado en el arreglo.

La figura 6 ilustra un arreglo 70 para almacenar parámetros de ALF y valores de índice de capa temporal (Templdx), de acuerdo con la segunda técnica de esta divulgación. En el ejemplo de la figura 6, el número de capas temporales posibles es 5 y el número máximo de conjuntos de filtros para la predicción temporal de una región se establece en 6, dando como resultado un arreglo 70 que contiene 30 entradas. En el ejemplo de la figura 6, suponiendo que cada imagen de la figura 3 está codificada con ALF habilitado y los filtros de ALF para cada imagen están explícitamente señalizados, antes de decodificar POC6 de la figura 3, los arreglos para los filtros de ALF almacenados tienen los estados que se muestran en la figura 6.

En el ejemplo de la figura 6, un codificador de vídeo puede revisar los valores Templdx almacenados en el arreglo 70 para determinar cuáles de las entradas en el arreglo 70 almacenan parámetros de ALF que el codificador de vídeo puede usar como predictores de los parámetros de ALF usados en la codificación de POC6. Al hacerlo, el codificador de vídeo puede ignorar cualquier entrada que especifique valores Templdx mayores que el  $T_3$  (es decir, el Templdx para POC6). A diferencia del ejemplo de la figura 4B, los filtros para POCO no se sobrescriben mediante los filtros para POC3.

De esta forma, de acuerdo con la segunda técnica de la invención reivindicada, el codificador de vídeo genera un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo. Una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual. Adicionalmente, el codificador de vídeo 20 reconstruye la imagen actual. El codificador de vídeo 20 también almacena, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Además, el codificador de vídeo 20 almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF. Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. El codificador de vídeo 20 determina, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. El codificador de vídeo 20 luego aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual. Después de aplicar el filtrado de bucle adaptativo a la región actual, el codificador de vídeo 20 puede usar la región actual para la predicción de una imagen subsecuente de los datos de vídeo.

De manera similar, de acuerdo con la segunda técnica de la invención reivindicada, el decodificador de vídeo 30 recibe un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo. Una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual. Además, el decodificador de vídeo 30 reconstruye la imagen actual. El decodificador de vídeo 30 almacena, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Adicionalmente, el decodificador de vídeo 30 almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF. Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. El decodificador de vídeo 30 determina, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. Adicionalmente, el decodificador de vídeo 30 aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual.

En algunos ejemplos basados en la segunda técnica de esta divulgación, también se puede registrar el valor de POC asociado con cada conjunto de filtros de ALF. Por ejemplo, un codificador de vídeo también puede almacenar, en un arreglo (por ejemplo, arreglo 70), un valor de POC de una imagen actual cuyos parámetros de ALF están codificados explícitamente. De este modo, en este ejemplo, después de codificar/decodificar una pluralidad de imágenes múltiples, el codificador de vídeo ha almacenado, en el arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de una nueva imagen actual.

En un ejemplo, los posibles candidatos para la predicción temporal pueden decidirse recorriendo los conjuntos incluidos en el arreglo, todos o algunos conjuntos de filtros con Templdx igual o menor y aquellos con un valor de POC igual a un valor de POC de una de las imágenes de referencia en una lista de imágenes de referencia actual pueden tratarse como candidatos efectivos. Por ejemplo, un codificador de vídeo puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado

indica la capa temporal a la cual pertenece una región actual de una imagen actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En este ejemplo, se requiere que, al determinar el conjunto aplicable de parámetros de ALF para la región actual, un valor de POC asociado con el conjunto aplicable de parámetros de ALF para la región actual sea igual a un valor de POC de una imagen de referencia en una lista de imágenes de referencia de la imagen actual.

En algunos ejemplos basados en la segunda técnica de esta divulgación, cada filtro de ALF que va a ser almacenado en el arreglo se asociará con una imagen de referencia que esté incluida en un conjunto de imágenes de referencia de la imagen actual (estas imágenes también estarían disponibles en el búfer de imágenes decodificadas). Es decir, si una imagen no está incluida en el conjunto de imágenes de referencia de la imagen actual, los filtros asociados con la imagen actual no se pueden almacenar y usar para la predicción temporal de ALF.

En algunos ejemplos basados en la segunda técnica de esta divulgación, el tamaño del arreglo puede depender del tamaño de un conjunto de imágenes de referencia. Por ejemplo, el tamaño del arreglo puede ser igual a un número máximo de imágenes de referencia que se permite que estén en un conjunto de imágenes de referencia.

En algunos ejemplos, un codificador de vídeo no genera una lista específicamente para los parámetros de filtro de ALF (es decir, parámetros de ALF), sino que la lista es la misma que las listas de imágenes de referencia que se generan para la sección actual. En este caso, los parámetros de ALF asociados con las imágenes de referencia de la región actual se almacenan directamente junto con otra información (tales como muestras de reconstrucción, información de movimiento de cada bloque con una región) requerida por el almacenamiento de imágenes de referencia. Como otra alternativa, la lista de parámetros de filtro de ALF se establece igual al conjunto de imágenes de referencia de la sección (o imagen) actual.

En otro ejemplo donde cada filtro de ALF almacenado en el arreglo (por ejemplo, arreglo 70) está asociado con una imagen de referencia incluida en el conjunto de imágenes de referencia de la imagen actual, la lista (arreglo) de parámetros de filtro de ALF (asociados con imágenes de referencia incluidas en el conjunto de imágenes de referencia de la imagen actual) se genera por separado independientemente de las listas de imágenes de referencia para la sección actual. Para la generación eficiente de una lista eficiente de parámetros de filtro de ALF, de tal manera que los conjuntos de parámetros de filtro de ALF usados con mayor frecuencia estén en posiciones anteriores en la lista de parámetros de filtro de ALF, los elementos de sintaxis para la señalización de un orden particular de los conjuntos candidatos de parámetro de filtro de ALF en la lista de parámetros de filtro de ALF se pueden incluir en un encabezado de sección, de manera similar como la sintaxis de modificación de lista de imágenes de referencia en el encabezado de sección.

De acuerdo con una tercera técnica, en lugar de usar la regla de FIFO para actualizar unos arreglos de filtros de ALF almacenados, se propone considerar además las diferencias de Conteo de Orden de Imagen (POC) para actualizar los arreglos. Por ejemplo, si un arreglo (por ejemplo, arreglo 50 de la figura 4A y la figura 4B, uno de los arreglos 60 de la figura 5, o el arreglo 70 de la figura 6) está lleno, un codificador de vídeo puede determinar qué entrada en el arreglo almacena filtros de ALF asociados con un valor de POC más diferente de un valor de POC de una imagen actual. En un ejemplo basado en la primera técnica, cuando se señala explícitamente un conjunto de parámetros de ALF para una región de una imagen actual, un codificador de vídeo puede determinar, con base en las diferencias entre un valor de POC de la imagen actual y los valores de POC asociados con conjuntos de parámetros de ALF, qué conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual reemplazar con el conjunto aplicable de parámetros de ALF para la región actual. En un ejemplo basado en la segunda técnica, cuando un conjunto de parámetros de ALF se señalan explícitamente para una región de una imagen actual, un codificador de vídeo puede determinar, con base en las diferencias entre un valor de POC de la imagen actual y los valores de POC asociados con conjuntos de parámetros de ALF, qué conjunto de parámetros de ALF en el arreglo reemplazar con el conjunto aplicable de parámetros de ALF para la región actual.

En algunos ejemplos, se puede definir una lista separada de filtros de selección a partir de un conjunto de imágenes de referencia que sea diferente de la selección de imágenes de referencia a partir de un conjunto de imágenes de referencia. En este caso, los filtros seleccionados pueden ser de una imagen que no esté incluida en ninguna lista de imágenes de referencia de la sección/mosaico/imagen actual.

De acuerdo con una cuarta técnica, la señalización de un índice de un conjunto/subconjunto seleccionado de filtros para la predicción temporal de ALF puede depender de un índice de capa temporal. Un subconjunto de filtros para la predicción temporal de ALF es un conjunto parcial de filtros de ALF. Por ejemplo, actualmente, podría haber 25 filtros de ALF por imagen. En este ejemplo, cuando se usa la predicción temporal, el codificador de vídeo puede elegir 10 en lugar de 25 filtros de ALF para asociarlos con una imagen. En un ejemplo, se puede usar el método de binarización unaria truncada para codificar el índice seleccionado de un conjunto de filtros y el valor máximo del número permitido de conjuntos es dependiente del índice de capa temporal.

Por ejemplo, de acuerdo con un ejemplo de la cuarta técnica, el codificador de vídeo 20 puede incluir, en un flujo de bits, un elemento de sintaxis que indica un índice de un conjunto seleccionado de parámetros de ALF. De manera similar, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, un elemento de sintaxis que indica un índice de un conjunto seleccionado de parámetros de ALF. El conjunto seleccionado de parámetros de ALF puede estar en uno de los arreglos de un tipo usado en la primera técnica o el arreglo de un tipo usado en la segunda técnica. En este ejemplo, el codificador de vídeo 20 y/o decodificador de vídeo 30 pueden determinar, con base en el conjunto seleccionado de parámetros de ALF en el arreglo, un conjunto aplicable de parámetros de ALF para la región actual. El codificador de vídeo 20 y/o decodificador de vídeo 30 pueden aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, un filtro de ALF a la región actual. En este ejemplo, un formato del elemento de sintaxis es dependiente de un índice de capa temporal. Por ejemplo, se puede usar un método de binarización unaria truncada para codificar el elemento de sintaxis y un valor máximo del número permitido de conjuntos de parámetros de ALF es dependiente del índice de capa temporal.

En algunos ejemplos basados en la cuarta técnica, la señalización del índice puede depender además de las diferencias de POC. En otras palabras, en el contexto del ejemplo del párrafo previo, el formato del elemento de sintaxis es dependiente además de las diferencias de POC. Por ejemplo, si el índice es 0, el conjunto seleccionado de parámetros de ALF se asocia con la imagen con un valor de POC más cercano a un valor de POC de una imagen actual; si el índice es 1, el conjunto seleccionado de parámetros de ALF se asocia con la imagen con un valor de POC más cercano al valor de POC de la imagen actual, y así sucesivamente. En este ejemplo, si dos o más de los conjuntos de parámetros de ALF en el arreglo o arreglos se asocian con imágenes que tienen la misma distancia de POC desde la imagen actual, los conjuntos de parámetros de ALF asociados con imágenes con valores de POC más bajos (o, en otros ejemplos, más altos) se asocian con valores de índice más bajos.

De acuerdo con una quinta técnica, en lugar de heredar tanto los coeficientes de filtro como la información de fusión de clases, se propone que solo se pueda heredar la información de fusión de clases. Es decir, los índices de filtro para diferentes clases podrían heredarse de la información codificada previamente. Alternativamente, además, se pueden asignar arreglos separados con un arreglo para registrar los índices de filtro para cada clase y el otro para registrar los coeficientes de filtro.

De este modo, en un ejemplo de acuerdo con la quinta técnica, un codificador de vídeo puede almacenar, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual, correspondiendo cada arreglo respectivo de la pluralidad de arreglos a una respectiva capa temporal diferente. En este ejemplo, el codificador de vídeo puede determinar, a partir del conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, información de fusión de clases y no coeficientes de filtro.

Además, en algunos ejemplos, el codificador de vídeo puede almacenar, en una segunda pluralidad de arreglos, conjuntos de coeficientes de filtro usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de una imagen actual, correspondiendo cada arreglo respectivo de la segunda pluralidad de arreglos a una capa temporal diferente respectiva. Como parte de la determinación del conjunto aplicable de parámetros de ALF para la región actual, el codificador de vídeo puede determinar, con base en un conjunto de coeficientes de filtro en uno de los arreglos de la segunda pluralidad de arreglos que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual y con base en el conjunto de parámetros de ALF en un arreglo en la primera pluralidad de arreglos que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, el conjunto aplicable de parámetros de ALF.

En un ejemplo de acuerdo con la quinta técnica, un codificador de vídeo puede almacenar, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. En este ejemplo, el codificador de vídeo puede determinar, a partir del conjunto de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, información de fusión de clases y no coeficientes de filtro.

Además, en algunos ejemplos, el codificador de vídeo puede almacenar, en un segundo arreglo, conjuntos de coeficientes de filtro usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. En tales ejemplos, el codificador de vídeo puede almacenar, en el segundo arreglo, índices de capa temporal asociados con los conjuntos de coeficientes de filtro. Un índice de capa temporal asociado con un conjunto de coeficientes de filtro indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. Como parte de la determinación del conjunto aplicable de parámetros de ALF para la región actual, el codificador de vídeo puede determinar, con base en un conjunto de coeficientes de filtro en uno de los arreglos de la segunda pluralidad de arreglos

que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual y con base en el conjunto de parámetros de ALF en un arreglo en la primera pluralidad de arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, el conjunto aplicable de parámetros de ALF para la región actual.

De acuerdo con una sexta técnica, en lugar de heredar tanto los coeficientes de filtro como la información de fusión de clases, se propone que solo se puedan heredar los coeficientes de filtro. Es decir, para la sección/imagen actual, la relación entre el índice de clase y el índice de filtro se puede señalar además incluso si se usa una predicción temporal.

De este modo, de acuerdo con un ejemplo de la sexta técnica, un codificador de vídeo puede almacenar, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. En este ejemplo, como parte de la determinación de un conjunto aplicable de parámetros de ALF para la región actual, el codificador de vídeo puede determinar, a partir del conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, coeficientes de filtro y no información de fusión de clases.

Además, en algunos ejemplos, el codificador de vídeo puede almacenar, en una segunda pluralidad de arreglos, conjuntos de información de fusión de clases usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Cada arreglo respectivo de la segunda pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Como parte de la determinación del conjunto aplicable de parámetros de ALF para la región actual, el codificador de vídeo puede determinar, con base en un conjunto de información de fusión de clases en uno de los arreglos de la segunda pluralidad de arreglos que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual y con base en el conjunto de parámetros de ALF en un arreglo en la primera pluralidad de arreglos que corresponde a la capa temporal a la cual pertenece la región actual o que corresponde a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, el conjunto aplicable de parámetros de ALF.

De acuerdo con otro ejemplo de la sexta técnica, un codificador de vídeo puede almacenar, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. En este ejemplo, el codificador de vídeo puede determinar, a partir del conjunto de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, coeficientes de filtro y no información de fusión de clases.

Además, en algunos ejemplos, el codificador de vídeo puede almacenar, en un segundo arreglo, conjuntos de información de fusión de clases usada en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. En este ejemplo, el codificador de vídeo también puede almacenar, en el segundo arreglo, índices de capa temporal asociados con los conjuntos de información de fusión de clases. Un índice de capa temporal asociado con un conjunto de coeficientes de filtro indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. Como parte de la determinación del conjunto aplicable de parámetros de ALF para la región actual, el codificador de vídeo puede determinar, con base en un conjunto de coeficientes de filtro en uno de los arreglos de la segunda pluralidad de arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual y con base en el conjunto de parámetros de ALF en un arreglo en la primera pluralidad de arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, el conjunto aplicable de parámetros de ALF para la región actual.

De acuerdo con una séptima técnica, incluso cuando se usa predicción temporal, las diferencias entre los filtros almacenados seleccionados y el filtro derivado pueden ser además señalizadas. En un ejemplo, el diseño actual para la señalización de predicción temporal que permite el indicador e índice de un conjunto de filtros todavía puede usarse. Además, puede usarse un indicador para indicar si señalizar o no las diferencias de filtro. Si sí, las diferencias pueden ser además señalizadas. En algunos ejemplos, pueden agregarse filtros de marcos o secciones codificadas previamente y tratarse como parte de filtros fijos. En este caso, el tamaño de filtros fijos y los coeficientes de los filtros fijos pueden cambiarse de manera adaptativa. Alternativamente, además, en algunos ejemplos, cuando se agrega un filtro de un conjunto a los filtros fijos, se aplicará una poda para evitar duplicaciones.

En un ejemplo de acuerdo con la séptima técnica, el codificador de vídeo puede determinar, con base en un conjunto seleccionado de parámetros de ALF en un arreglo que corresponde a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. Alternativamente,

en este ejemplo, el codificador de vídeo 20 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En cualquier caso, el codificador de vídeo 20 puede incluir, en el flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual. En algunos ejemplos, el codificador de vídeo 20 puede incluir, en el flujo de bits, un elemento de sintaxis que indica si el flujo de bits incluye la indicación de la diferencia.

En otro ejemplo de acuerdo con la séptima técnica, el decodificador de vídeo 30 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en un arreglo que corresponde a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. Alternativamente, en este ejemplo, el decodificador de vídeo 30 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En cualquier caso, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual. En este ejemplo, como parte de la determinación del conjunto aplicable de parámetros de ALF para la región actual, el decodificador de vídeo 30 puede determinar, con base en el conjunto seleccionado de parámetros de ALF y la diferencia, el conjunto aplicable de parámetros de ALF para la región actual. En algunos ejemplos, el decodificador de vídeo 30 puede obtener, del flujo de bits, un elemento de sintaxis que indica si el flujo de bits incluye la indicación de la diferencia.

De acuerdo con una octava técnica, uno o más conjuntos de filtros de ALF pueden almacenarse en conjuntos de parámetros (por ejemplo, conjuntos de parámetros de secuencia o conjuntos de parámetros de imagen) de tal manera que las imágenes incluso en diferentes secuencias de vídeo codificadas puedan usarlos. Para evitar problemas de resiliencia a errores o problemas de acceso aleatorio, no se permite actualizar los conjuntos de filtros de ALF en conjuntos de parámetros usando filtros de ALF señalados en encabezados de secciones. Por ejemplo, al codificar un flujo de bits, un codificador de vídeo puede almacenar, en un arreglo, conjuntos de filtros de ALF especificados en un conjunto de parámetros del flujo de bits. En este ejemplo, los encabezados de secciones pueden incluir parámetros de ALF que definen filtros de ALF adicionales o diferencias de filtros. Un encabezado de sección es una parte de una sección codificada (o segmento de sección codificada) que contiene los elementos de datos que pertenecen a la primera o todas las unidades de árbol de codificación representadas en la sección (o segmento de sección).

La figura 7 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo 20 que puede implementar las técnicas de esta divulgación. La figura 7 se proporciona con propósitos de explicación y no debe considerarse limitante de las técnicas que se ejemplifican y describen ampliamente en esta divulgación. Las técnicas de esta divulgación pueden ser aplicables a diversos estándares o métodos de codificación.

La circuitería de procesamiento incluye un codificador de vídeo 20, y el codificador de vídeo 20 está configurado para realizar una o más de las técnicas de ejemplo descritas en esta divulgación. Por ejemplo, el codificador de vídeo 20 incluye una circuitería integrada, y las diversas unidades ilustradas en la figura 5 pueden formarse como bloques de circuitos de hardware que están interconectados con un bus de circuito. Estos bloques de circuitos de hardware pueden ser bloques de circuitos separados o dos o más de las unidades pueden combinarse en un bloque de circuitos de hardware común. Los bloques de circuitos de hardware pueden formarse como una combinación de componentes eléctricos que forman bloques de operaciones tales como unidades lógicas aritméticas (ALU), unidades de funciones elementales (EFU), así como bloques lógicos tales como AND, OR, NAND, NOR, XOR, XNOR y otros bloques lógicos similares.

En algunos ejemplos, una o más de las unidades ilustradas en la figura 7 pueden ser unidades de software que se ejecutan en la circuitería de procesamiento. En tales ejemplos, el código objeto para estas unidades de software se almacena en la memoria. Un sistema operativo puede hacer que el codificador de vídeo 20 recupere el código objeto y ejecute el código objeto, lo cual hace que el codificador de vídeo 20 realice operaciones para implementar las técnicas de ejemplo. En algunos ejemplos, las unidades de software pueden ser firmware que el codificador de vídeo 20 ejecuta en el inicio. Por consiguiente, el codificador de vídeo 20 es un componente estructural que tiene hardware que realiza las técnicas de ejemplo o tiene software/firmware que se ejecuta en el hardware para especializar el hardware para realizar las técnicas de ejemplo.

En el ejemplo de la figura 7, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 100, memoria de datos de vídeo 101, 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 procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, un búfer de imágenes decodificadas 116 y una unidad de codificación por entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestra).

La memoria de datos de vídeo 101 puede estar configurada para almacenar datos de vídeo que van a ser codificados mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 101 pueden obtenerse, por ejemplo, de la fuente de vídeo 18. El búfer de imágenes decodificadas 116 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en modos de intra o intercodificación. La memoria de datos de vídeo 101 y el búfer de imágenes decodificadas 116 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, tal como memoria de acceso aleatorio dinámica (DRAM), incluyendo DRAM sincrónica (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 101 y el búfer de imágenes decodificadas 116 pueden estar proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 101 puede estar en el chip con otros componentes del codificador de vídeo 20, o fuera del chip en relación con esos componentes. La memoria de datos de vídeo 101 puede ser la misma que o parte del medio de almacenamiento 19 de la figura 1.

El codificador de vídeo 20 recibe datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en una sección de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada con bloques de árbol de codificación (CTB) de luma igualmente dimensionados y los CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar división para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado con una CTU de acuerdo con una estructura de árbol.

El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 100 puede dividir los bloques de codificación asociados con la CU entre una o más PU de la CU. De este modo, cada PU puede estar asociada con un bloque de predicción de luma y bloques de predicción de croma correspondientes. El codificador de vídeo 20 y decodificador de vídeo 30 pueden soportar PU que tengan diversos tamaños. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es  $2N \times 2N$ , el codificador de vídeo 20 y el decodificador de vídeo 30 pueden soportar tamaños de PU de  $2N \times 2N$  o  $N \times N$  para intrapredicción, y tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  o similares para interpredicción. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden soportar división asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$  para interpredicción.

La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU. Como parte de la generación de los datos predictivos para una PU, la unidad de procesamiento de interpredicción 120 realiza una interpredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de interpredicción 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en una sección I, una sección P o una sección B. En una sección I, todas las PU se intrapredicen. Por tanto, si la PU está en una sección I, la unidad de procesamiento de interpredicción 120 no realiza una interpredicción en la PU. De este modo, para los bloques codificados en modo I, el bloque predicho se forma usando la predicción espacial de los bloques vecinos codificados previamente dentro del mismo marco. Si una PU está en una sección P, la unidad de procesamiento de interpredicción 120 puede usar una interpredicción unidireccional para generar un bloque predictivo de la PU. Si una PU está en una sección B, la unidad de procesamiento de interpredicción 120 puede usar interpredicción unidireccional o bidireccional para generar un bloque predictivo de la PU.

La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 126 puede realizar una intrapredicción en las PU en secciones I, secciones P y secciones B.

Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de intrapredicción 126 puede usar muestras de bloques de muestra de PU vecinas para generar un bloque predictivo para una PU. Las PU vecinas pueden estar por encima, por encima y a la derecha, por encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccionales. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada con la PU.

La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpredicción 120 para las PU o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos

ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU con base en métricas de tasa/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en este documento como los bloques predictivos seleccionados.

5

La unidad de generación residual 102 puede generar, con base en los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para una CU y los bloques predictivos seleccionados (por ejemplo, bloques predictivos de luma, Cb y Cr) para las PU de la CU, bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de tal manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque predictivo seleccionado correspondiente de una PU de la CU.

10

La unidad de procesamiento de transformada 104 puede dividir los bloques residuales de una CU en bloques de transformada de las TU de la CU. Por ejemplo, la unidad de procesamiento de transformada 104 puede realizar una división de árbol cuádruple para dividir los bloques residuales de la CU en bloques de transformada de las TU de la CU. De este modo, una TU puede estar asociada con un bloque de transformada de luma y dos bloques de transformada de croma. Los tamaños y posiciones de los bloques de transformada de luma y croma de las TU de una CU pueden o pueden no basarse en los tamaños y las posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuádruple conocida como un "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos de hoja del RQT.

15

20

La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado con 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 a un bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En tales ejemplos, el bloque de transformada puede tratarse como un bloque de coeficientes de transformada.

25

30

La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse hacia abajo a un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado con una TU de una CU con base en un valor de parámetro de cuantificación (QP) asociado con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con una CU ajustando el valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información. De este modo, los coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

35

40

La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede agregar muestras del bloque residual reconstruido a muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformada reconstruido asociado con una TU. Al reconstruir bloques de transformada para cada TU de una CU de esta forma, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

45

50

La unidad de filtro 114 puede realizar una o más operaciones de eliminación de bloque para reducir los artefactos que generan bloques en los bloques de codificación asociados con una CU. La unidad de filtro 114 puede realizar las técnicas de filtro de esta divulgación. Por ejemplo, la unidad de filtro 114 puede almacenar, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. En este ejemplo, cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Además, en este ejemplo, la unidad de filtro 114 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En este ejemplo, la unidad de filtro 114 puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a uno o más bloques en la región actual.

55

60

En otro ejemplo, la unidad de filtro 114 puede almacenar, en un arreglo (por ejemplo, arreglo 70 de la figura 6), conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual. Adicionalmente, la unidad de filtro 114 puede almacenar, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF. Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual

65



fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF. Además, en este ejemplo, la unidad de filtro 114 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En este ejemplo, la unidad de filtro 114 puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a uno o más bloques en la región actual.

El búfer de imágenes decodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones de eliminación de bloque en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos en el búfer de imágenes decodificadas 116 para realizar una intrapredicción en otras PU en la misma imagen que la CU.

La unidad de codificación por entropía 118 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 118 puede recibir bloques de coeficientes de la unidad de cuantificación 106 y puede recibir elementos de sintaxis de la unidad de procesamiento de predicción 100. La unidad de codificación por entropía 118 puede realizar una o más operaciones de codificación por entropía sobre los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación de CABAC, una operación de codificación de longitud variable adaptativa al contexto (CAVLC), 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 por Entropía de División de Intervalos de Probabilidad (PIPE), una operación de codificación de Golomb Exponencial u otro tipo de operación de codificación por entropía sobre los datos. El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos codificados por entropía generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan valores de coeficientes de transformada para una CU.

La figura 8 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo 30 que está configurado para implementar las técnicas de esta divulgación. La figura 8 se proporciona con propósitos de explicación y no es limitante de las técnicas que se ejemplifican y describen ampliamente en esta divulgación. Con propósitos de explicación, esta divulgación describe el decodificador de vídeo 30 en el contexto de la codificación de HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o métodos de codificación.

La circuitería de procesamiento incluye un decodificador de vídeo 30, y el decodificador de vídeo 30 está configurado para realizar una o más de las técnicas de ejemplo descritas en esta divulgación. Por ejemplo, el decodificador de vídeo 30 incluye circuitería integrada, y las diversas unidades ilustradas en la figura 8 pueden formarse como bloques de circuito de hardware que están interconectados con un bus de circuito. Estos bloques de circuito de hardware pueden ser bloques de circuito separados o dos o más de las unidades pueden combinarse en un bloque de circuito de hardware común. Los bloques de circuito de hardware pueden formarse como una combinación de componentes eléctricos que forman bloques de operación tales como unidades lógicas aritméticas (ALU), unidades de función elemental (EFU), así como bloques lógicos tales como AND, OR, NAND, NOR, XOR, XNOR y otros bloques lógicos similares.

En algunos ejemplos, una o más de las unidades ilustradas en la figura 8 pueden ser unidades de software que se ejecutan en la circuitería de procesamiento. En tales ejemplos, el código objeto para estas unidades de software se almacena en la memoria. Un sistema operativo puede hacer que el decodificador de vídeo 30 recupere el código objeto y ejecute el código objeto, lo cual hace que el decodificador de vídeo 30 realice operaciones para implementar las técnicas de ejemplo. En algunos ejemplos, las unidades de software pueden ser firmware que el decodificador de vídeo 30 ejecuta en el inicio. Por consiguiente, el decodificador de vídeo 30 es un componente estructural que tiene hardware que realiza las técnicas de ejemplo o tiene software/firmware que se ejecuta en el hardware para especializar el hardware para realizar las técnicas de ejemplo.

En el ejemplo de la figura 8, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 150, memoria de datos de vídeo 151, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y un búfer de imágenes decodificadas 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. En otros ejemplos, el decodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

La memoria de datos de vídeo 151 puede almacenar datos de vídeo codificados, tales como un flujo de bits de vídeo codificado, que van a ser decodificados mediante los componentes de decodificador de vídeo 30. Los

datos de vídeo almacenados en la memoria de datos de vídeo 151 pueden obtenerse, por ejemplo, a partir de un medio legible por ordenador 16, por ejemplo, de una fuente de vídeo local, tal como una cámara, a través de una comunicación por red cableada o inalámbrica de datos de vídeo, o accediendo a un medio de almacenamiento de datos físicos. La memoria de datos de vídeo 151 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. El búfer de imágenes decodificadas 162 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para uso en la decodificación de datos de vídeo por el decodificador de vídeo 30, por ejemplo, en modos de intra o intercodificación, o para la salida. La memoria de datos de vídeo 151 y el búfer de imágenes decodificadas 162 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (DRAM), incluyendo DRAM sincrónica (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 151 y el búfer de imágenes decodificadas 162 pueden estar proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 151 puede estar en el chip con otros componentes del decodificador de vídeo 30, o fuera del chip en relación con esos componentes. La memoria de datos de vídeo 151 puede ser la misma que o parte del medio de almacenamiento 28 de la figura 1.

La memoria de datos de vídeo 151 recibe y almacena datos de vídeo codificados (por ejemplo, unidades de NAL) de un flujo de bits. La unidad de decodificación por entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades de NAL) de la memoria de datos de vídeo 151 y puede analizar sintácticamente las unidades de NAL para obtener elementos de sintaxis. La unidad de decodificación por entropía 150 puede decodificar por entropía elementos de sintaxis codificados por entropía en las unidades de NAL. La unidad de procesamiento de predicción 152, unidad de cuantificación inversa 154, unidad de procesamiento de transformada inversa 156, unidad de reconstrucción 158 y unidad de filtro 160 pueden generar datos de vídeo decodificados con base en los elementos de sintaxis extraídos del flujo de bits. La unidad de decodificación por entropía 150 puede realizar un proceso generalmente recíproco al de la unidad de codificación por entropía 118.

Además de obtener elementos de sintaxis del flujo de bits, el decodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU, el decodificador 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 decodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar de manera inversa, es decir, descuantificar, bloques de coeficientes asociados con la TU. Después de que la unidad de cuantificación inversa 154 cuantifique de manera inversa un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes con el fin de generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada Karhunen-Loeve (KLT) inversa, una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

La unidad de cuantificación inversa 154 puede realizar técnicas particulares de esta divulgación. Por ejemplo, para al menos un grupo de cuantificación respectivo de una pluralidad de grupos de cuantificación dentro de un CTB de una CTU de una imagen de los datos de vídeo, la unidad de cuantificación inversa 154 puede derivar, con base al menos en parte en la información de cuantificación local señalizada en el flujo de bits, un parámetro de cuantificación respectivo para el grupo de cuantificación respectivo. Adicionalmente, en este ejemplo, la unidad de cuantificación inversa 154 puede cuantificar de manera inversa, con base en el parámetro de cuantificación respectivo para el grupo de cuantificación respectivo, al menos un coeficiente de transformada de un bloque de transformada de una TU de una CU de la CTU. En este ejemplo, el grupo de cuantificación respectivo se define como un grupo de CU o bloques de codificación sucesivos, en orden de codificación, de tal manera que los límites del grupo de cuantificación respectivo deben ser límites de las CU o bloques de codificación y un tamaño del grupo de cuantificación respectivo es mayor que o igual a un umbral. El decodificador de vídeo 30 (por ejemplo, unidad de procesamiento de transformada inversa 156, unidad de reconstrucción 158 y unidad de filtro 160) puede reconstruir, con base en coeficientes de transformada cuantificados inversos del bloque de transformada, un bloque de codificación de la CU.

Si una PU se codifica usando intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar intrapredicción para generar bloques predictivos de la PU. La unidad de procesamiento de intrapredicción 166 puede usar un modo de intrapredicción para generar los bloques predictivos de la PU con base en muestras de bloques espacialmente vecinos. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU con base en uno o más elementos de sintaxis obtenidos del flujo de bits.

Si una PU se codifica usando interpredicción, la unidad de decodificación por entropía 150 puede determinar información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, con

base en la información de movimiento de la PU, uno o más bloques de referencia. La unidad de compensación de movimiento 164 puede generar, con base en el uno o más bloques de referencia, bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para la PU.

5 La unidad de reconstrucción 158 puede usar bloques de transformada (por ejemplo, bloques de transformada de luma, Cb y Cr) para las TU de una CU y los bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) de las PU de la CU, es decir, ya sea datos de intrapredicción o datos de interpredicción, según sea aplicable, para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de reconstrucción 158 puede agregar muestras de los bloques de transformada (por ejemplo, bloques de transformada de luma, Cb y Cr) a las muestras correspondientes de los bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU.

15 La unidad de filtro 160 puede realizar una operación de eliminación de bloque para reducir los artefactos que generan bloques asociados con los bloques de codificación de la CU. La unidad de filtro 160 puede realizar las técnicas de filtro de esta divulgación. Por ejemplo, la unidad de filtro 160 puede almacenar, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de una imagen actual. En este ejemplo, cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Por ejemplo, para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, la unidad de filtro 20 160 puede almacenar, en el arreglo respectivo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo. En este ejemplo, la unidad de filtro 160 puede determinar, 25 con base en un conjunto seleccionado de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. Además, en este ejemplo, la unidad de filtro 160 puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, un filtro de ALF a uno o más bloques en la región actual.

30 En otro ejemplo, la unidad de filtro 160 almacena, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de una imagen actual. Además, en este ejemplo, la unidad de filtro 160 almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF. Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de 35 parámetros de ALF para aplicar un filtro de ALF. En este ejemplo, la unidad de filtro 160 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual. En este ejemplo, la unidad de filtro 160 puede aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, un filtro de ALF a uno o más bloques de la región actual.

40 El decodificador de vídeo 30 puede almacenar los bloques de codificación de la CU en el búfer de imágenes decodificadas 162. El búfer de imágenes decodificadas 162 puede proporcionar imágenes de referencia para la compensación de movimiento subsecuente, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1. Por ejemplo, el decodificador de vídeo 30 puede 45 realizar, con base en los bloques en el búfer de imágenes decodificadas 162, operaciones de intrapredicción o inter para las PU de otras CU.

Se han descrito ciertos aspectos de esta divulgación con respecto a extensiones del estándar de HEVC con propósitos de ilustración. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros 50 procesos de codificación de vídeo, incluyendo otros procesos de codificación de vídeo estándar o registrados aún no desarrollados.

La figura 9 es un diagrama de flujo que ilustra una operación de ejemplo del codificador de vídeo 20, de acuerdo con la primera técnica de esta divulgación. Los diagramas de flujo de esta divulgación se proporcionan como 55 ejemplos. En otros ejemplos, las acciones pueden realizarse en diferentes órdenes, o las operaciones pueden incluir más, menos o diferentes acciones.

En el ejemplo de la figura 9, el codificador de vídeo 20 genera un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo (200). Una región actual (por ejemplo, una sección actual u otra unidad) de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual 60 pertenece la región actual. El codificador de vídeo 20 puede generar el flujo de bits de acuerdo con cualquiera de los ejemplos descritos en otras partes en esta divulgación, tal como el ejemplo de la figura 5.

Adicionalmente, el codificador de vídeo 20 reconstruye la imagen actual (202). Por ejemplo, el codificador de 65 vídeo 20 puede reconstruir un bloque de la imagen actual agregando muestras de bloques residuales reconstruidos a muestras correspondientes de uno o más bloques predictivos para producir bloques

reconstruidos. Al reconstruir bloques de esta forma, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la imagen actual.

Además, para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, el codificador de vídeo 20 puede almacenar, en el arreglo respectivo (por ejemplo, uno de los arreglos 60 de la figura 5), conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo (204). Cada conjunto de parámetros de ALF puede incluir un conjunto de coeficientes de filtro y/o un conjunto de información de fusión de clases de ALF.

El codificador de vídeo 20 determina, con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual (206). Por ejemplo, el codificador de vídeo 20 puede seleccionar el conjunto seleccionado de parámetros de ALF usando un análisis de distorsión de tasa de los conjuntos de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual. En algunos ejemplos, el conjunto aplicable de parámetros de ALF para la región actual puede ser el mismo que el conjunto seleccionado de parámetros de ALF. En algunos ejemplos, el codificador de vídeo 20 puede incluir, en el flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual.

El codificador de vídeo 20 puede entonces aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual (208). Al aplicar el filtrado de bucle adaptativo a la región actual, el codificador de vídeo 20 puede aplicar un filtro de ALF a uno o más, pero no necesariamente a todos, los bloques en la región actual. Por ejemplo, el codificador de vídeo 20 puede dividir la región actual en bloques (por ejemplo, bloques 4x4). En este ejemplo, para cada uno de los bloques, el codificador de vídeo 20 puede determinar (por ejemplo, con base en una dirección y actividad del bloque) una categoría correspondiente para el bloque. En este ejemplo, el conjunto aplicable de parámetros de ALF para la región actual puede incluir coeficientes de filtro de un filtro de ALF de la categoría para el bloque. En este ejemplo, el codificador de vídeo 20 puede entonces aplicar el filtro de ALF al bloque.

Después de aplicar el filtrado de bucle adaptativo a la región actual, el codificador de vídeo 20 usa la región actual para la predicción de una imagen subsecuente de los datos de vídeo (210). Por ejemplo, el codificador de vídeo 20 puede usar la región actual para la interpredicción de un bloque de la imagen subsecuente.

La figura 10 es un diagrama de flujo que ilustra una operación de ejemplo del decodificador de vídeo 30, de acuerdo con la primera técnica de esta divulgación. En el ejemplo de la figura 10, el decodificador de vídeo 30 recibe un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo (250). Una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual.

Además, el decodificador de vídeo 30 reconstruye la imagen actual (252). El decodificador de vídeo 30 puede reconstruir la imagen actual de acuerdo con cualquiera de los ejemplos proporcionados en otras partes en esta divulgación. Por ejemplo, el decodificador de vídeo 30 puede reconstruir un bloque de la imagen actual agregando muestras de bloques residuales reconstruidos a muestras correspondientes de uno o más bloques predictivos para producir bloques reconstruidos. Al reconstruir bloques de esta forma, el decodificador de vídeo 30 puede reconstruir los bloques de codificación de la imagen actual.

El decodificador de vídeo 30 también almacena, en una pluralidad de arreglos, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual (254). Cada arreglo respectivo de la pluralidad de arreglos corresponde a una capa temporal diferente respectiva. Por ejemplo, para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, el decodificador de vídeo 30 puede almacenar, en el arreglo respectivo (por ejemplo, uno de los arreglos 60 de la figura 5), conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo.

Adicionalmente, el decodificador de vídeo 30 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual (256). Por ejemplo, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, un índice que indica el conjunto seleccionado de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual. En algunos ejemplos, el conjunto aplicable de parámetros de ALF para la región actual puede ser el mismo que el conjunto seleccionado de parámetros de ALF. En algunos ejemplos, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual.

El decodificador de vídeo 30 puede entonces aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual (258). Al aplicar el filtrado de bucle adaptativo a la región actual, el decodificador de vídeo 30 puede aplicar un filtro de ALF a uno o más, pero no necesariamente a todos, los bloques en la región actual. Por ejemplo, el decodificador de vídeo 30 puede dividir la región actual en bloques (por ejemplo, bloques 4x4). En este ejemplo, para cada uno de los bloques, el decodificador de vídeo 30 puede determinar (por ejemplo, con base en una dirección y actividad del bloque) una categoría correspondiente para el bloque. En este ejemplo, el conjunto aplicable de parámetros de ALF para la región actual puede incluir coeficientes de filtro de un filtro de ALF de la categoría para el bloque. En este ejemplo, el decodificador de vídeo 30 puede entonces aplicar el filtro de ALF al bloque.

La figura 11 es un diagrama de flujo que ilustra una operación de ejemplo del codificador de vídeo 20 de acuerdo con la segunda técnica de esta divulgación. En el ejemplo de la figura 11, el codificador de vídeo 20 genera un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo (300). Una región actual (por ejemplo, una sección actual u otra unidad) de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual. El codificador de vídeo 20 puede generar el flujo de bits de acuerdo con cualquiera de los ejemplos descritos en otras partes en esta divulgación, tal como el ejemplo de la figura 5.

Además, el codificador de vídeo 20 puede reconstruir la imagen actual (302). Por ejemplo, el codificador de vídeo 20 puede reconstruir un bloque de la imagen actual agregando muestras de bloques residuales reconstruidos a muestras correspondientes de uno o más bloques predictivos para producir bloques reconstruidos. Al reconstruir bloques de esta forma, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la imagen actual.

El codificador de vídeo 20 almacena, en un arreglo (por ejemplo, el arreglo 70 de la figura 6), conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual (304). Adicionalmente, el codificador de vídeo 20 almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF (306). Un índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF.

Además, el codificador de vídeo 20 determina, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual o una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual (308). Por ejemplo, el codificador de vídeo 20 puede seleccionar el conjunto seleccionado de parámetros de ALF usando un análisis de distorsión de tasa de los conjuntos de parámetros de ALF en el arreglo. En algunos ejemplos, el conjunto aplicable de parámetros de ALF para la región actual puede ser el mismo que el conjunto seleccionado de parámetros de ALF. En algunos ejemplos, el codificador de vídeo 20 puede incluir, en el flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual.

El codificador de vídeo 20 aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, un ALF a la región actual (310). El codificador de vídeo 20 puede aplicar el filtro de ALF a la región actual de acuerdo con cualquiera de los ejemplos proporcionados en otras partes en esta divulgación.

Después de aplicar el filtro de ALF a la región actual, el codificador de vídeo 20 usa la región actual para la predicción de una imagen subsecuente de los datos de vídeo (312). Por ejemplo, el codificador de vídeo 20 puede usar la región actual para la interpredicción de un bloque de la imagen subsecuente.

La figura 12 es un diagrama de flujo que ilustra una operación de ejemplo del decodificador de vídeo 30 de acuerdo con una técnica de la divulgación. En el ejemplo de la figura 12, el decodificador de vídeo 30 recibe un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo (350). Una región actual (por ejemplo, una sección actual u otra unidad) de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual.

El decodificador de vídeo 30 puede entonces reconstruir la imagen actual (352). El decodificador de vídeo 30 puede reconstruir la imagen actual de acuerdo con cualquiera de los ejemplos proporcionados en otras partes en esta divulgación. Por ejemplo, el decodificador de vídeo 30 puede reconstruir un bloque de la imagen actual agregando muestras de bloques residuales reconstruidos a muestras correspondientes de uno o más bloques predictivos para producir bloques reconstruidos. Al reconstruir bloques de esta forma, el decodificador de vídeo 30 puede reconstruir los bloques de codificación de la imagen actual.

En el ejemplo de la figura 12, el decodificador de vídeo 30 almacena, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual (354). Además, el decodificador de vídeo 30 almacena, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF (356). Un índice de capa temporal asociado con

un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF.

5 El decodificador de vídeo 30 puede determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual.

10 (358). Por ejemplo, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, un índice que indica el conjunto seleccionado de parámetros de ALF en el arreglo. En algunos ejemplos, el conjunto aplicable de parámetros de ALF para la región actual puede ser el mismo que el conjunto seleccionado de parámetros de ALF. En algunos ejemplos, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual.

15 El decodificador de vídeo 30 luego aplica, con base en el conjunto aplicable de parámetros de ALF para la región actual, un filtro de ALF a la región actual (360). El decodificador de vídeo 30 puede aplicar el filtro de ALF a la región actual de acuerdo con cualquiera de los ejemplos proporcionados en otras partes en esta divulgación.

20 Un codificador de vídeo, como se describe en esta divulgación, puede hacer referencia a un codificador de vídeo o a un decodificador de vídeo. De manera similar, una unidad de codificación de vídeo puede hacer referencia a un codificador de vídeo o a un decodificador de vídeo. Asimismo, la codificación de vídeo puede hacer referencia a la codificación de vídeo o a la decodificación de vídeo, según sea aplicable. En esta divulgación, la expresión "basado en" puede indicar basado solo en, basado al menos en parte en o basado de alguna forma en. Esta divulgación puede usar el término "unidad de vídeo" o "bloque de vídeo" o "bloque" para hacer referencia a uno o más bloques de muestra y estructuras de sintaxis usadas para codificar muestras del uno o más bloques de muestras. Los tipos de ejemplo de unidades de vídeo pueden incluir CTU, CU, PU, unidades de transformada (TU), macrobloques, divisiones de macrobloque, y así sucesivamente. En algunos contextos, la discusión de PU puede intercambiarse con la discusión de macrobloques o divisiones de macrobloque. Los tipos de ejemplo de bloques de vídeo pueden incluir bloques de árbol de codificación, bloques de codificación y otros tipos de bloques de datos de vídeo.

35 Las técnicas de esta divulgación se pueden aplicar a la codificación de vídeo en soporte de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión satelital, transmisiones de vídeo de generación de flujo por Internet, tales como generación de flujo adaptativa dinámica sobre HTTP (DASH), vídeo digital que está codificado en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

40 Se debe reconocer que dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en este documento se pueden realizar en una secuencia diferente, se pueden agregar, fusionar o excluir por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos se pueden realizar de manera concurrente, por ejemplo, a través de procesamiento multiproceso, procesamiento de interrupción o múltiples procesadores, en lugar de secuencialmente.

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

65 A modo de ejemplo, y sin limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disquete óptico, almacenamiento en disquete magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar el código de programa deseado en la forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. También, cualquier conexión se denomina adecuadamente un medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio

web, servidor u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea de suscriptor digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que en cambio se dirigen a medios de almacenamiento tangibles, no transitorios. Disquete y disco, como se usan en este documento, incluyen disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete flexible y disco Blu-ray, donde los disquetes usualmente reproducen datos de manera magnética, mientras que los discos reproducen datos de manera óptica con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

La funcionalidad descrita en esta divulgación puede ser realizada por circuitería de procesamiento programable y/o de función fija. Por ejemplo, las instrucciones pueden ser ejecutadas por circuitería de procesamiento programable y/o de función fija. Tal circuitería de procesamiento puede incluir uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), arreglos lógicos programables en campo (FPGA) u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término "procesador", como se usa en este documento puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en este documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. También, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos. Los circuitos de procesamiento pueden estar acoplados a otros componentes de diversas formas. Por ejemplo, un circuito de procesamiento puede estar acoplado a otros componentes a través de una interconexión de dispositivo interno, una conexión de red cableada o inalámbrica u otro medio de comunicación.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un aparato telefónico inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización mediante diferentes unidades de hardware. En cambio, como se describió anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describió anteriormente, en conjunto con software y/o firmware adecuados.

Se han descrito diversos ejemplos.

## REIVINDICACIONES

1. Un método para decodificar datos de vídeo, comprendiendo el método:

5 recibir (250; 350) un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo, en donde una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual;

reconstruir (252; 352) la imagen actual;

10 ya sea:

para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, almacenar (254), en el arreglo respectivo, conjuntos de parámetros de filtrado de bucle adaptativo, ALF, usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo; y

15 determinar (256), con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

o

25 almacenar (354), en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual y almacenar (356), en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF, en donde el índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF; y

determinar (358), con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

35 y

aplicar (258; 360), con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual.

40

2. Un método de codificación de datos de vídeo, comprendiendo el método:

generar (200; 300) un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo, en donde una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual;

45

reconstruir (202; 302) la imagen actual;

ya sea:

50

para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, almacenar (204), en el arreglo respectivo, conjuntos de parámetros de filtrado de bucle adaptativo, ALF, usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo; y

55

determinar (206), con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual o arreglos de la pluralidad de arreglos que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

60

o:

almacenar (304), en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual y almacenar (306), en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF, en donde el índice de

65



capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual el conjunto de parámetros de ALF fue usado para aplicar un filtro de ALF; y

5 determinar (308), con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

10 aplicar (208;310), con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual; y

después de aplicar el filtrado de bucle adaptativo a la región actual, usar (210;312) la región actual para la predicción de una imagen subsecuente de los datos de vídeo.

15 3. El método de la reivindicación 1 o 2, en donde, cuando los parámetros de ALF se almacenan en la pluralidad de arreglos, al menos dos de la pluralidad de arreglos incluyen diferentes números de conjuntos de parámetros de ALF.

20 4. El método de la reivindicación 1 o 2, en donde almacenar el conjunto aplicable de parámetros de ALF para la región actual comprende determinar, con base en las diferencias entre un valor de Conteo de Orden de Imagen, POC, de la imagen actual y los valores de POC asociados con conjuntos de parámetros de ALF, qué conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual reemplazar con el conjunto aplicable de parámetros de ALF para la región actual.

25 5. El método de la reivindicación 1 o 2, en donde se requiere que, al determinar el conjunto aplicable de parámetros de ALF para la región actual, un valor de Conteo de Orden de Imagen, POC, asociado con el conjunto aplicable de parámetros de ALF para la región actual sea igual a un valor de POC de una imagen de referencia en una lista de imágenes de referencia de la imagen actual.

30 6. El método de la reivindicación 1, que comprende además:

obtener, a partir del flujo de bits, un elemento de sintaxis que indica un índice del conjunto seleccionado de parámetros de ALF,

35 en donde determinar el conjunto aplicable de parámetros de ALF para la región actual comprende determinar, con base en el elemento de sintaxis, el conjunto seleccionado de parámetros de ALF, y

en donde un formato del elemento de sintaxis es dependiente de un índice temporal que indica la capa temporal a la cual pertenece la región actual.

40 7. El método de la reivindicación 1 o 2, en donde cuando los parámetros de ALF se almacenan en la pluralidad de arreglos determinar el conjunto aplicable de parámetros de ALF para la región actual comprende determinar, a partir del conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual, información de fusión de clases y no coeficientes de filtro.

45 8. El método de la reivindicación 1 o 2, en donde cuando los parámetros de ALF se almacenan en la pluralidad de arreglos determinar el conjunto aplicable de parámetros de ALF para la región actual comprende determinar, a partir del conjunto de parámetros de ALF en el arreglo que corresponde a la capa temporal a la cual pertenece la región actual, coeficientes de filtro y no información de fusión de clases.

50 9. El método de la reivindicación 1, que comprende además:

obtener, a partir del flujo de bits, una indicación de una diferencia entre el conjunto seleccionado de parámetros de ALF y el conjunto aplicable de parámetros de ALF para la región actual,

55 en donde determinar el conjunto aplicable de parámetros de ALF para la región actual comprende determinar, con base en el conjunto seleccionado de parámetros de ALF y la diferencia, el conjunto aplicable de parámetros de ALF para la región actual.

60 10. Un decodificador de vídeo (30) configurado para:

recibir un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo, en donde una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual;

65 reconstruir la imagen actual;

ya sea:

para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, almacenar, en el arreglo respectivo, conjuntos de parámetros de filtrado de bucle adaptativo, ALF, usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo; y

determinar, con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual o que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

o:

almacenar, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual y almacenar, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF, en donde el índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF; y

determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

y

aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual.

11. Un codificador de vídeo (20) configurado para:

generar un flujo de bits que incluye una representación codificada de una imagen actual de los datos de vídeo, en donde una región actual de la imagen actual está asociada con un índice temporal que indica una capa temporal a la cual pertenece la región actual;

reconstruir la imagen actual;

ya sea:

para cada arreglo respectivo de una pluralidad de arreglos que corresponden a diferentes capas temporales, almacenar, en el arreglo respectivo, conjuntos de parámetros de bucle adaptativo, ALF, usados en la aplicación de filtros de ALF a muestras de regiones de imágenes de los datos de vídeo que se decodifican antes de la región actual y que están en la capa temporal que corresponde al arreglo respectivo o en una capa temporal inferior a la capa temporal que corresponde al arreglo respectivo; y

determinar, con base en un conjunto seleccionado de parámetros de ALF en uno de los arreglos que corresponden a la capa temporal a la cual pertenece la región actual o arreglos de la pluralidad de arreglos que corresponden a una capa temporal inferior a la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

o

almacenar, en un arreglo, conjuntos de parámetros de ALF usados en la aplicación de filtros de ALF a muestras de imágenes de los datos de vídeo decodificados antes de la imagen actual y almacenar, en el arreglo, índices de capa temporal asociados con los conjuntos de parámetros de ALF, en donde el índice de capa temporal asociado con un conjunto de parámetros de ALF indica una capa temporal de una región en la cual fue usado el conjunto de parámetros de ALF para aplicar un filtro de ALF; y

determinar, con base en un conjunto seleccionado de parámetros de ALF en el arreglo cuyo índice de capa temporal asociado indica la capa temporal a la cual pertenece la región actual, un conjunto aplicable de parámetros de ALF para la región actual;

aplicar, con base en el conjunto aplicable de parámetros de ALF para la región actual, filtrado de bucle adaptativo a la región actual; y

después de aplicar el filtrado de bucle adaptativo a la región actual, usar la región actual para la predicción de una imagen subsecuente de los datos de vídeo.

12. Un medio de almacenamiento de datos legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un decodificador de vídeo lleven a cabo el método de cualquiera de las reivindicaciones 1 o 3 a 9.

5

13. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un codificador de vídeo lleven a cabo el método de cualquiera de las reivindicaciones 2 a 5, 7 u 8.

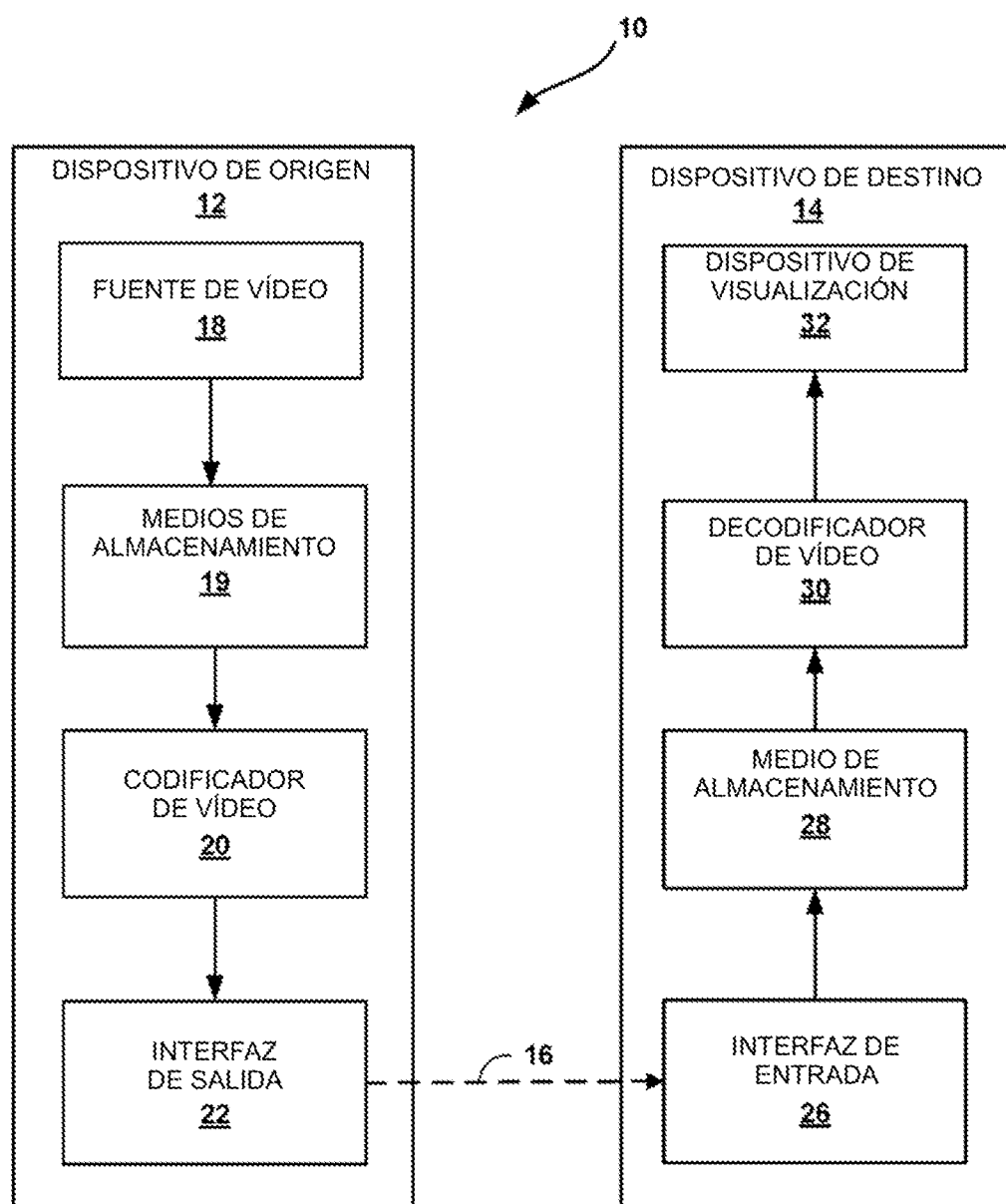


FIG. 1

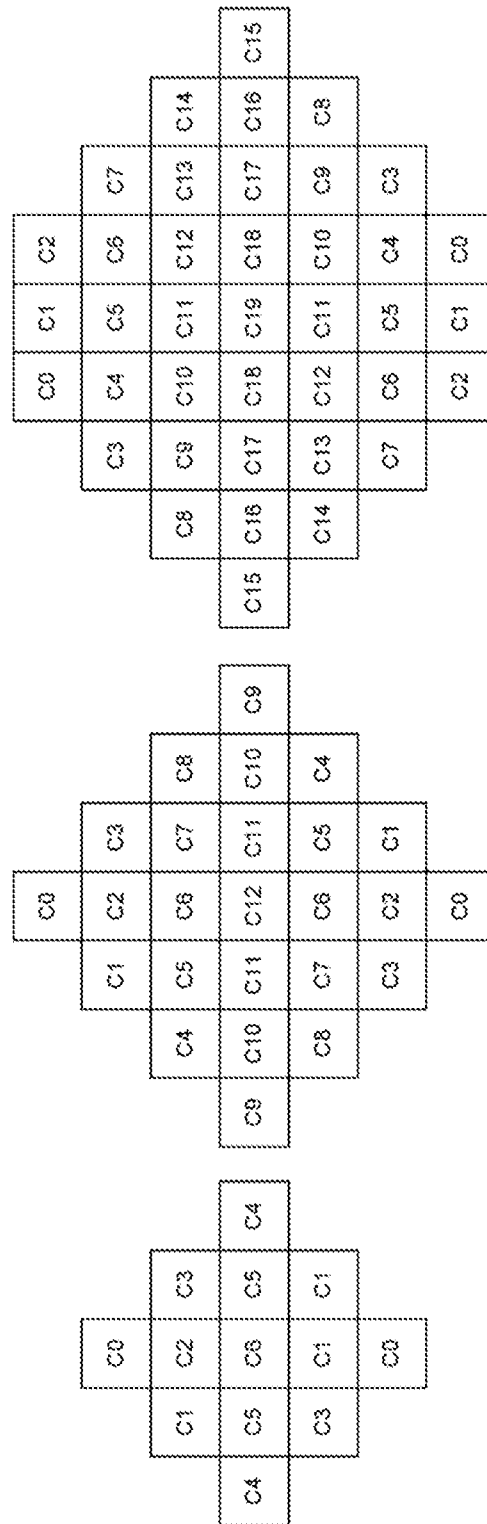


FIG. 2

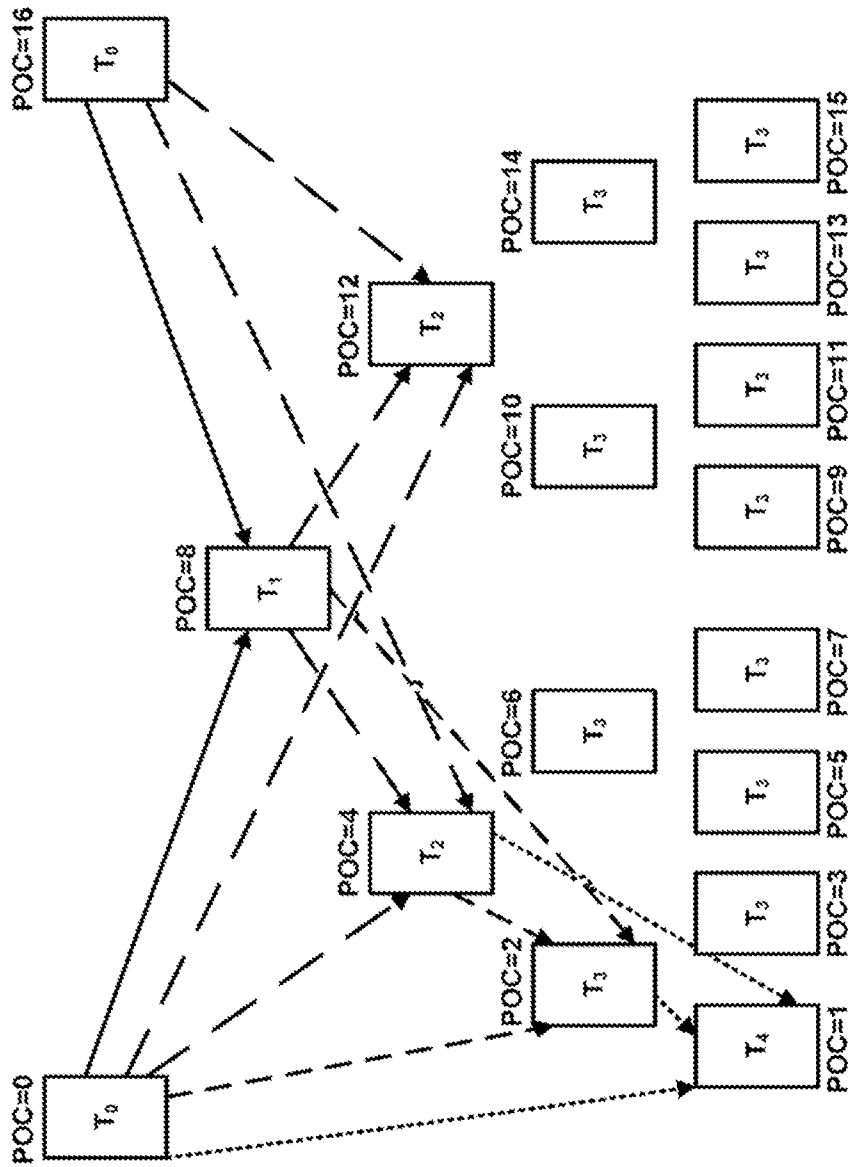


FIG. 3

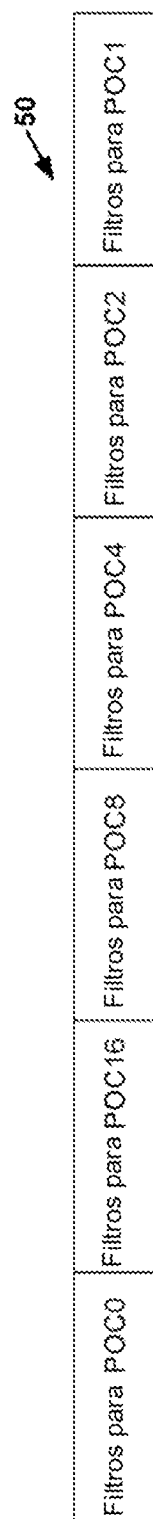


FIG. 4A

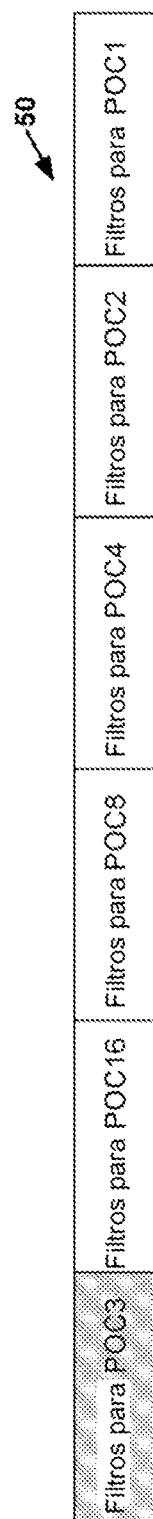


FIG. 4B

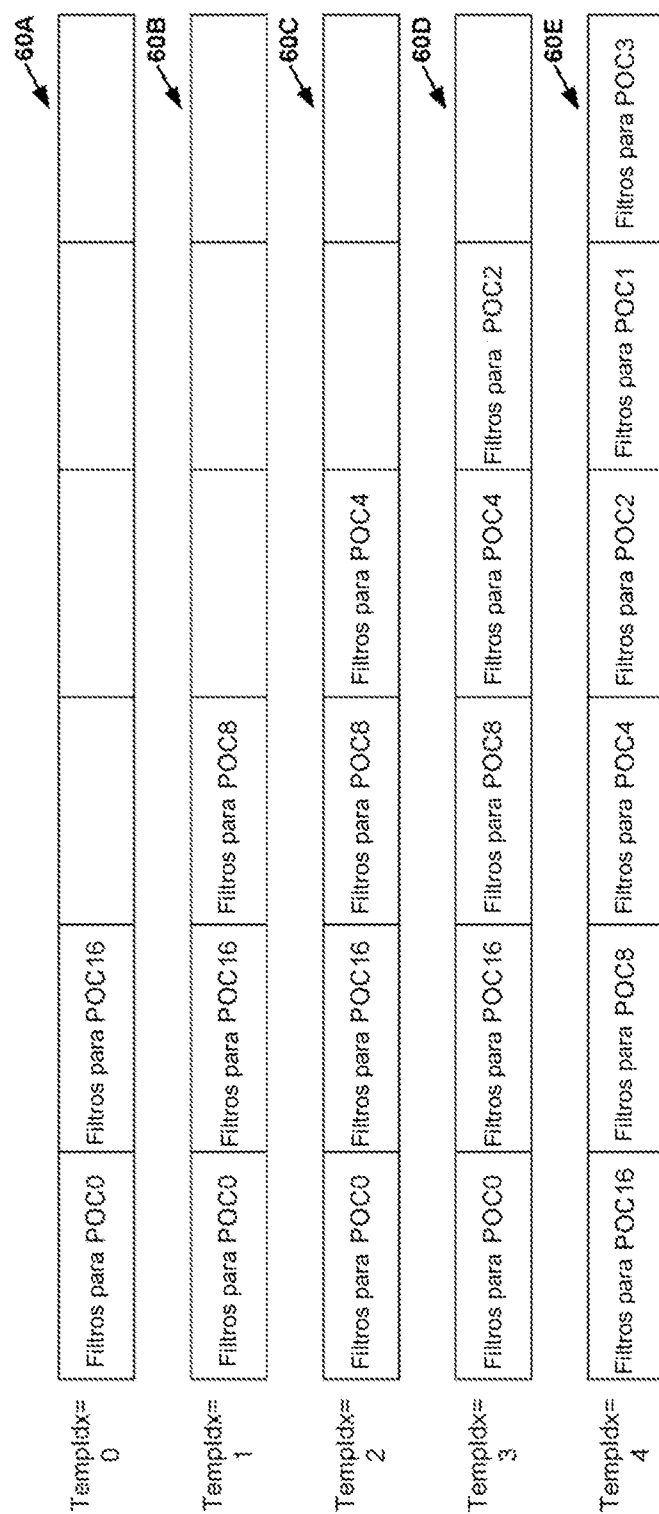


FIG. 5



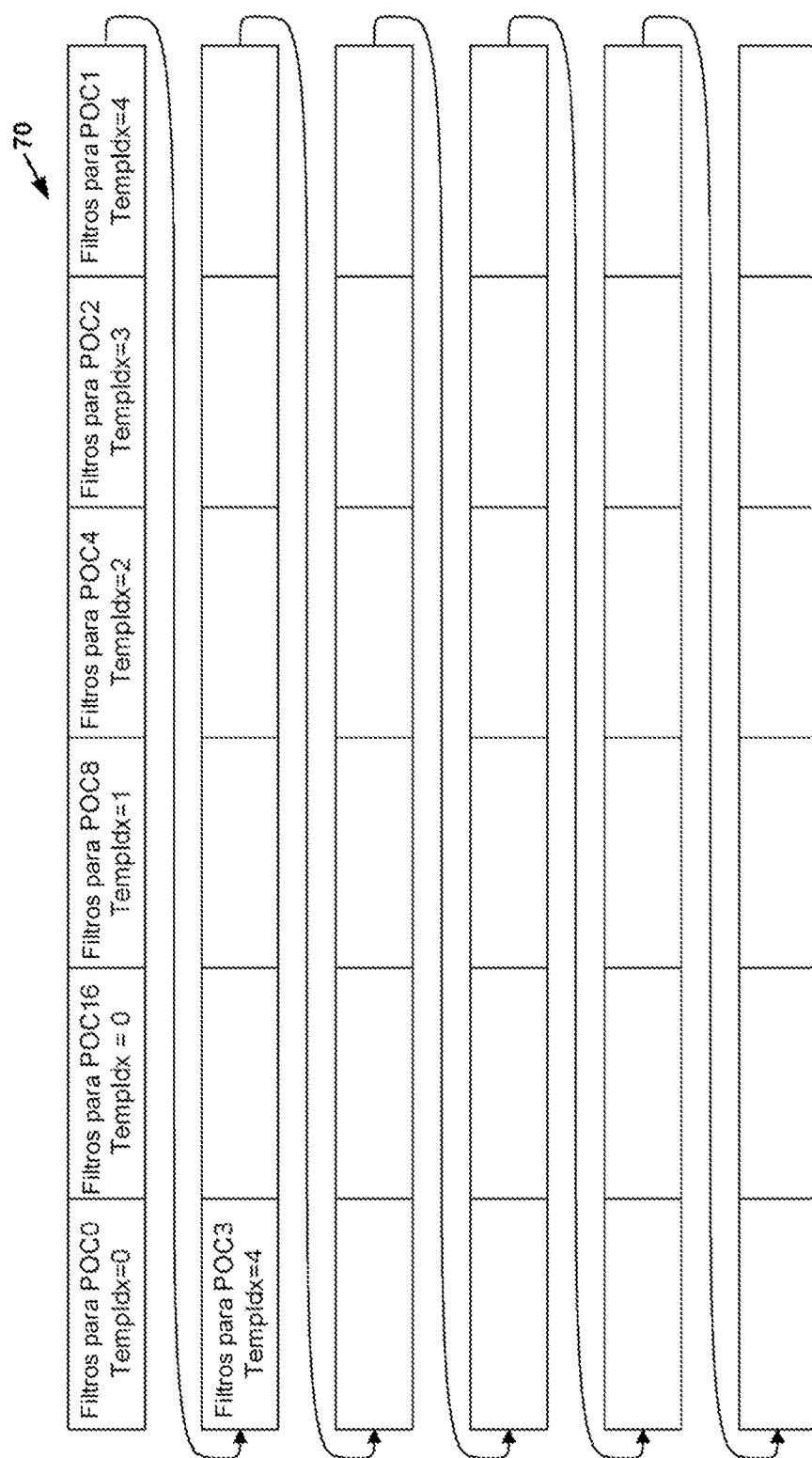


FIG. 6

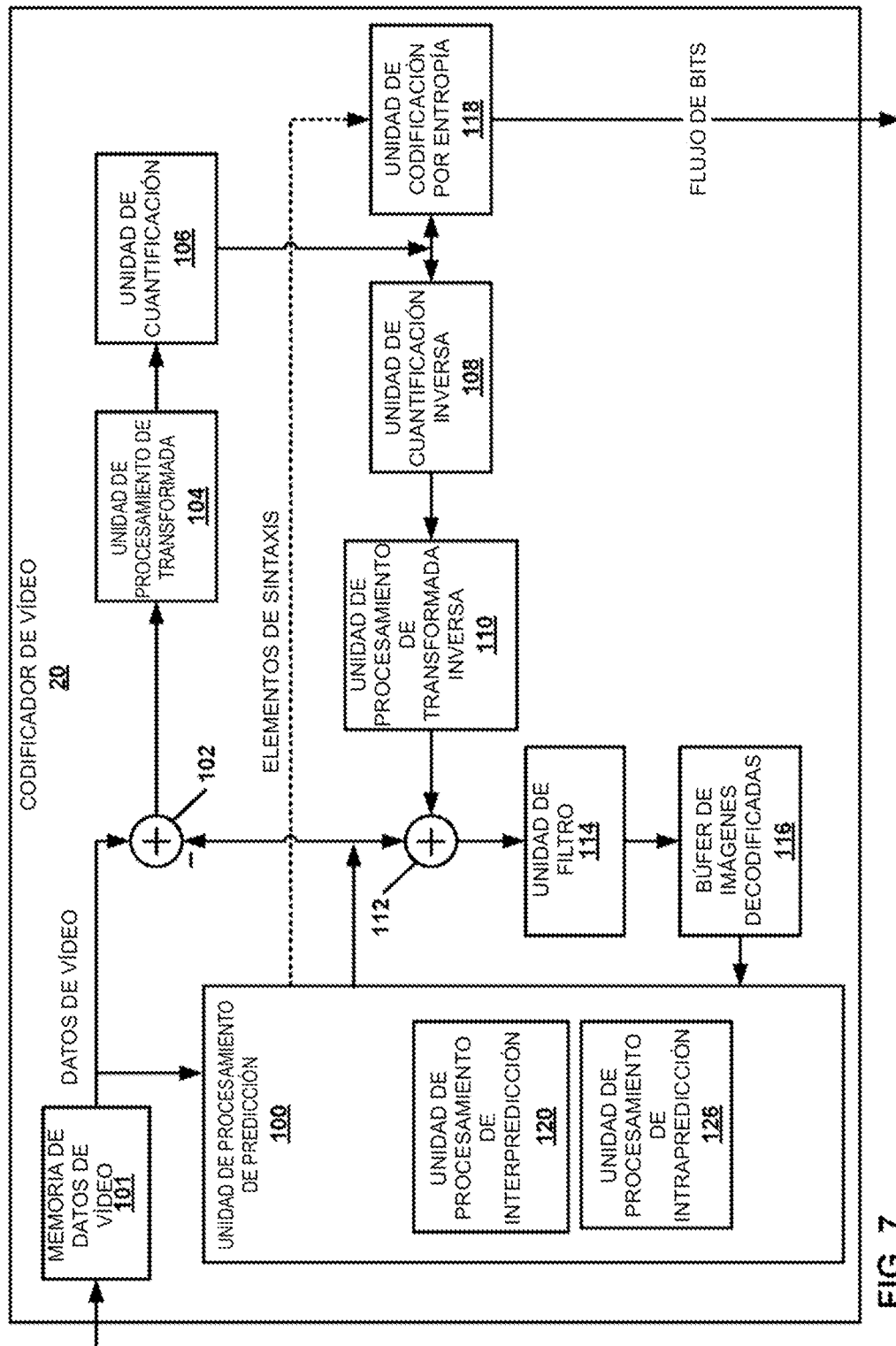


FIG. 7

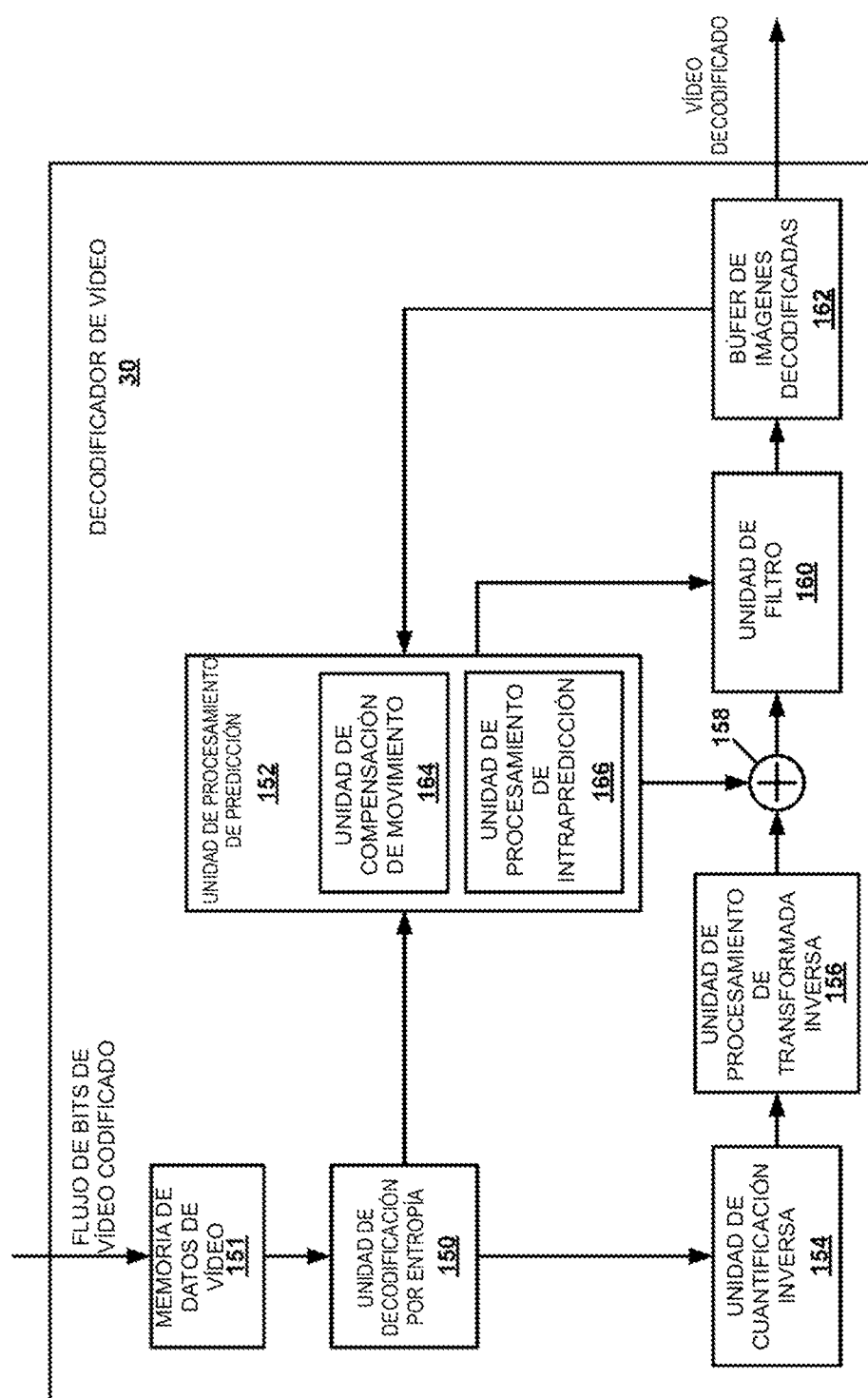


FIG. 8

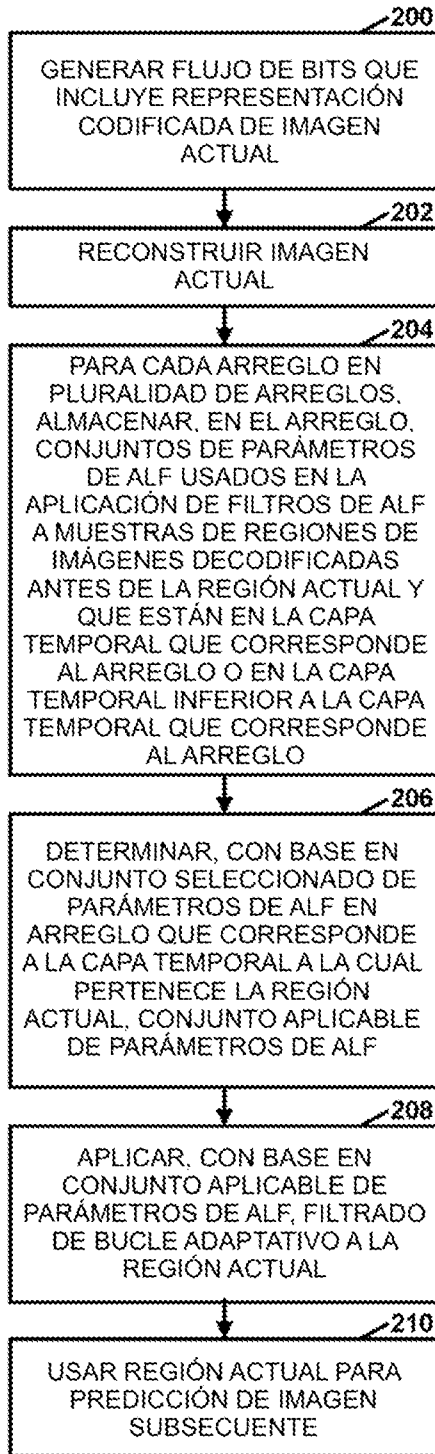


FIG. 9

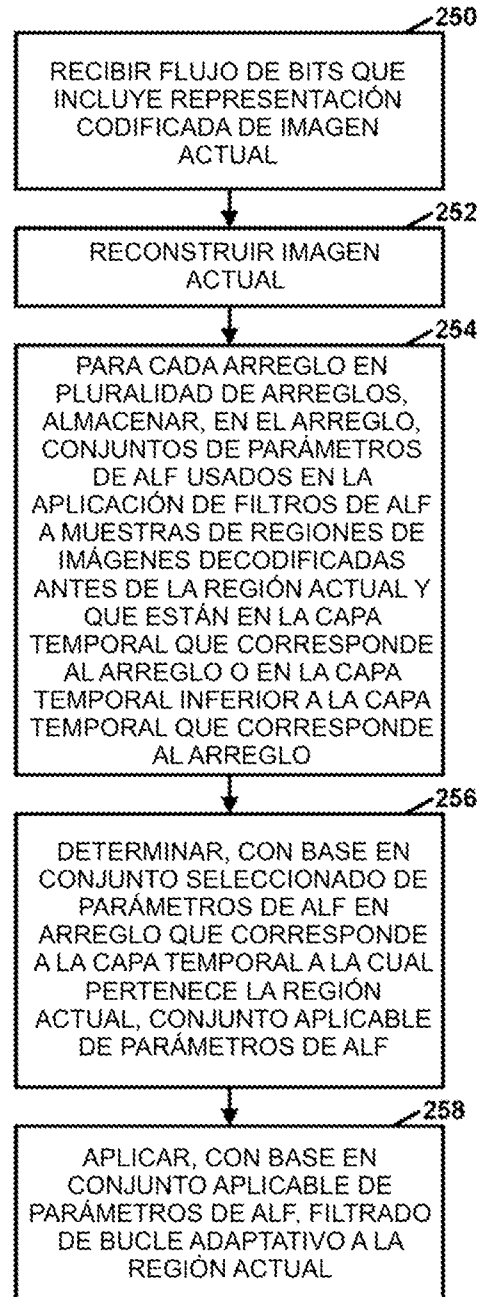


FIG. 10

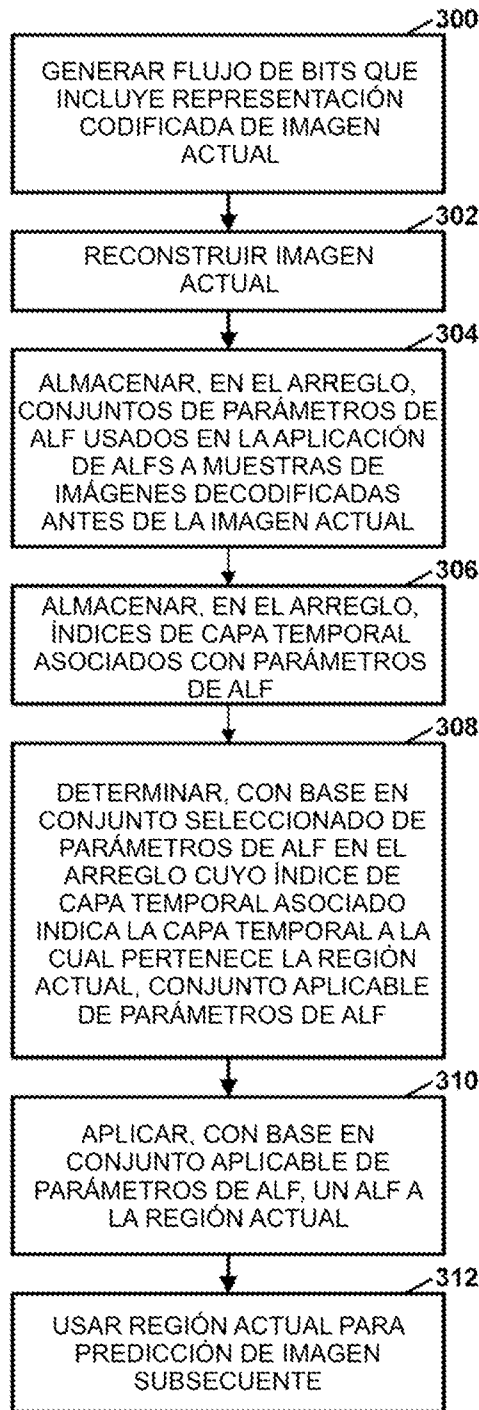


FIG. 11

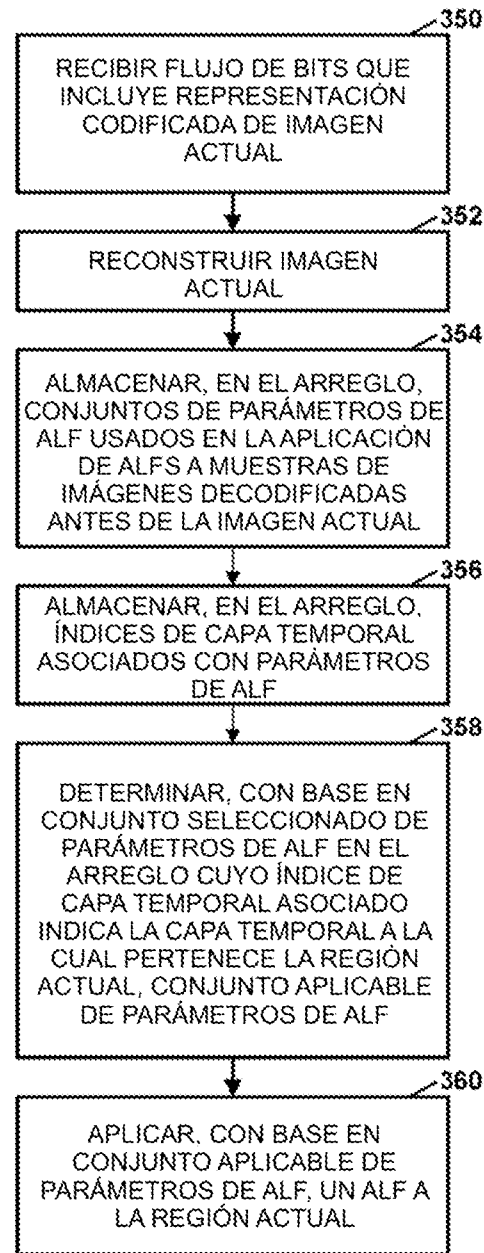


FIG. 12