

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 863 275**

51 Int. Cl.:

**H04N 19/117** (2014.01)  
**H04N 19/136** (2014.01)  
**H04N 19/176** (2014.01)  
**H04N 19/70** (2014.01)  
**H04N 19/82** (2014.01)  
**H04N 19/14** (2014.01)  
**H04N 19/463** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **15.02.2017** **PCT/US2017/017958**  
87 Fecha y número de publicación internacional: **24.08.2017** **WO17142939**  
96 Fecha de presentación y número de la solicitud europea: **15.02.2017** **E 17708062 (9)**  
97 Fecha y número de publicación de la concesión europea: **17.03.2021** **EP 3417613**

54 Título: **Transformaciones geométricas para filtros para la codificación de video**

30 Prioridad:

**15.02.2016 US 201662295461 P**  
**19.04.2016 US 201662324776 P**  
**14.02.2017 US 201715432839**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.10.2021**

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 863 275 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

Transformaciones geométricas para filtros para la codificación de video

5 Campo técnico

Esta divulgación se refiere a la codificación de video.

Antecedentes

10 Las capacidades de video digital pueden incorporarse en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radiotelefonos móviles o por satélite, los denominados "teléfonos inteligentes", dispositivos de video teleconferencia, dispositivos de transmisión en tiempo real de videos y similares. Los dispositivos de video digital implementan técnicas de compresión de video, tal 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 Video Avanzada (AVC), el estándar recientemente finalizado ITU-T H.265, Codificación de Video de Alta Eficiencia (HEVC), y extensiones de tales estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficiente implementando tales técnicas de compresión de video.

25 Las técnicas de compresión de video realizan predicciones espaciales (intraimagen) y/o predicciones temporales (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de video. Para la codificación de video con base en bloques, un segmento de video (es decir, un fotograma de video o una porción de un fotograma de video) puede dividirse en bloques de video, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia en los bloques vecinos en la misma imagen. Los bloques de video en un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse fotogramas y las imágenes de referencia pueden denominarse fotogramas de referencia.

35 La predicción espacial o temporal da como resultado un bloque predictivo para que se codifique un bloque. Los datos residuales representan las diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, dispuestos inicialmente en un arreglo bidimensional, pueden escanearse para producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr una compresión aún mayor.

45 El documento US 2011069752 divulga un procedimiento de codificación de imágenes en movimiento que puede generar información de borde que indica un atributo de un borde en una imagen decodificada local correspondiente a una imagen codificada. El procedimiento puede generar, además, con base en la información de borde, información de control asociada con la aplicación de un filtro a una imagen decodificada en un lado de decodificación.

50 El documento EP2584781 divulga un filtro adaptativo que incluye una sección de detección de bordes que discrimina la direccionalidad de una imagen de entrada en cada una de una pluralidad de regiones unitarias que constituyen la imagen de entrada.

55 El documento EP2048886 divulga un procedimiento para emplear un filtro de interpolación adaptativo. Las propiedades tal como las simetrías y otras limitaciones del filtro de interpolación adaptativo se establecen de antemano para controlar el número de coeficientes de filtro independientes.

60 Ehsan Maani y otros, en "Parametric Adaptive Loop Filter", 5. JCT-VC MEETING; 96. MPEG MEETING; 16-3-2011 - 23-3-2011; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), no. JCTVC-E320, ISSN 0000-0005 proporciona una breve descripción y los resultados del Filtro de Bucle Adaptativo Paramétrico y discute las formas de filtro con base en la dirección. Ver también Karcewicz M y otros, en "Study of coding efficiency improvements beyond HEVC", 113. MPEG MEETING; 19-10-2015 - 23-10-2015; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), (20151015), núm. m37102, y "Algorithm Description of Joint Exploration Test Model 1 (JEM1)", 113. MPEG MEETING; 19-10-2015 - 23-10-2015; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), (20151211), núm. N15790

## Sumario

En general, esta divulgación divulga técnicas relacionadas con el filtrado, tal como el filtrado de bucle adaptativo (ALF). En particular, varias técnicas para generar filtros para diferentes clases de bloques de datos de video pueden realizarse por un codificador de video, tal como un codificador de video o un decodificador de video. En un ejemplo, el codificador de video puede construir múltiples conjuntos de coeficientes de filtro e índices de código en los conjuntos de coeficientes de filtro para varias clases de bloques, donde cada índice identifica el conjunto de coeficientes de filtro para la clase de bloques correspondiente. En otro ejemplo, el codificador de video puede generar un filtro para una clase de bloques al usar un conjunto de coeficientes de filtro de un filtro generado previamente y aplicando una transformación geométrica (tal como una rotación, un volteo en vertical o un volteo en diagonal) a una región de soporte del filtro o a los propios coeficientes de filtro. En otro ejemplo más, el codificador de video puede predecir coeficientes de filtro para una clase de bloques a partir de un filtro fijo o un filtro de una imagen codificada previamente.

En las reivindicaciones independientes se define un procedimiento para filtrar un bloque decodificado de datos de video, un dispositivo para filtrar un bloque decodificado de datos de video.

Un medio de almacenamiento legible por ordenador (tal como un medio de almacenamiento no transitorio legible por ordenador) que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un procesador ejecute el procedimiento reivindicado también se define en las reivindicaciones independientes.

Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción a continuación. Otras características, objetivos y ventajas serán evidentes a partir de la descripción y los dibujos y a partir de las reivindicaciones.

### Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video ilustrativo que puede usar las técnicas descritas en esta divulgación.

La Figura 2 es un diagrama conceptual que ilustra un mapeo de intervalos para una métrica de actividad y una métrica de dirección para filtros.

La Figura 3 es un diagrama conceptual que ilustra un patrón ilustrativo para señalar diferencias de coeficientes de filtro.

La Figura 4 es un diagrama de bloques que ilustra un codificador de video ilustrativo que puede implementar las técnicas descritas en esta divulgación.

La Figura 5 es un diagrama de bloques que ilustra un decodificador de video ilustrativo que puede implementar las técnicas descritas en esta divulgación.

La Figura 6 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video).

La Figura 7 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video).

La Figura 8 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video).

La Figura 9 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video).

La Figura 10 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video).

La Figura 11 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video).

### Descripción detallada

En un codificador de video típico, el fotograma de una secuencia de video original se divide en regiones o bloques rectangulares, que se codifican en intramodo (modo I) o intermodo (modo P). Los bloques se codifican mediante el uso de algún tipo de codificación de transformación, tal como la codificación DCT. Sin embargo, la codificación pura con base en transformadas solo reduce la correlación entre píxeles dentro de un bloque particular, sin considerar la

correlación entre bloques de píxeles, y todavía produce altas tasas de transmisión de bits. Los estándares actuales de codificación de imágenes digitales también aprovechan ciertos procedimientos que reducen la correlación de valores de píxeles entre bloques. Como se aclarará en la siguiente descripción, el término codificación de video puede usarse en esta divulgación para referirse genéricamente a codificación de video o decodificación de video.

En general, los bloques codificados en modo P se predicen a partir de uno de los fotogramas previamente codificados y transmitidos. La información de predicción de un bloque está representada por un vector de movimiento bidimensional (2D). Para los bloques codificados en modo I, el bloque predicho se forma mediante el uso de la predicción espacial de bloques vecinos ya codificados dentro del mismo fotograma. El error de predicción, es decir, la diferencia entre el bloque que se codifica y el bloque predicho, se representa como un conjunto de funciones de base ponderadas de alguna transformada discreta. La transformación se realiza típicamente en bloques básicos de 8x8 o 4x4. Los pesos, coeficientes de transformación, se cuantifican subsecuentemente. La cuantificación introduce pérdida de información y, por lo tanto, los coeficientes cuantificados tienen menor precisión que los originales.

Los coeficientes de transformación cuantificados, junto con los vectores de movimiento y cierta información de control, forman una representación de secuencia codificada completa y se denominan elementos de sintaxis. Antes de la transmisión desde el codificador al decodificador, los elementos de sintaxis se codifican en entropía para reducir aún más el número de bits usados para representar los elementos de sintaxis.

En el decodificador, el bloque en el fotograma actual se obtiene construyendo primero su predicción en la misma manera que en el codificador y agregando a la predicción el error de predicción comprimido. El error de predicción comprimido se encuentra ponderando las funciones de la base de transformación mediante el uso de los coeficientes cuantificados. La diferencia entre el fotograma reconstruido y el fotograma original se denomina error de reconstrucción.

Para mejorar aún más la calidad del video decodificado, un decodificador de video puede realizar una o más operaciones de filtrado en los bloques de video reconstruidos. Ejemplos de estas operaciones de filtrado incluyen filtrado de desbloqueo, filtrado de desplazamiento adaptativo de muestra (SAO) y filtrado de bucle adaptativo (ALF). Los parámetros para estas operaciones de filtrado pueden ser determinados por un codificador de video y señalizados explícitamente en el flujo de bits de video codificado o pueden determinarse implícitamente por un decodificador de video.

Esta divulgación describe técnicas asociadas con el filtrado de datos de video reconstruidos en procesos de codificación y/o decodificación de video y, más particularmente, esta divulgación describe técnicas relacionadas con ALF. De acuerdo con esta divulgación, el filtrado se aplica en un codificador y la información del filtro se codifica en el flujo de bits para permitir que un decodificador identifique el filtrado que se aplicó en el codificador. El codificador de video puede probar varios escenarios de filtrado diferentes y, con base en un análisis de distorsión de velocidad, elegir un filtro o un conjunto de filtros que produzca un desplazamiento deseado entre la calidad del video reconstruido y la calidad de la compresión. El decodificador de video recibe datos de video codificados que incluyen la información del filtro, decodifica los datos de video y aplica un filtrado con base en la información de filtrado. De esta manera, el decodificador de video aplica el mismo filtrado que se aplicó en el codificador de video.

En esta divulgación, el término "filtro" generalmente se refiere a un conjunto de coeficientes de filtro. Por ejemplo, un filtro de 3x3 puede definirse mediante un conjunto de 9 coeficientes de filtro, un filtro de 5x5 puede definirse mediante un conjunto de 25 coeficientes de filtro, un filtro de 9x5 puede definirse mediante un conjunto de 45 coeficientes de filtro, etc. El término "conjunto de filtros" generalmente se refiere a un grupo de más de un filtro. Por ejemplo, un conjunto de dos filtros 3x3 podría incluir un primer conjunto de 9 coeficientes de filtro y un segundo conjunto de 9 coeficientes de filtro. El término "forma", a veces llamado "soporte de filtro", generalmente se refiere al número de hileras de coeficientes de filtro y al número de columnas de coeficientes de filtro para un filtro en particular. Por ejemplo, 9x9 es un ejemplo de una primera forma, 7x5 es un ejemplo de una segunda forma y 5x9 es un ejemplo de una tercera forma. En algunos casos, los filtros pueden adoptar formas no rectangulares que incluyen formas de diamante, formas similares a diamante, formas circulares, formas similares a círculos, formas hexagonales, formas octogonales, formas de cruz, formas de X, formas de T, otras formas geométricas, o muchas otras formas o configuraciones. En este caso, para los soportes de filtro no rectangulares, el número de coeficientes de filtro puede ser diferente. Por ejemplo, para un soporte de filtro de diamante de 9x9, el filtro de diamante de 9x9 puede definirse mediante un conjunto de coeficientes de filtro  $(9 \times 9 / 4 + 1)$ .

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video 10 ilustrativo que puede implementar las técnicas descritas en esta divulgación. Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo fuente 12 que genera datos de video codificados para decodificarse en un momento posterior por un dispositivo de destino 14. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores portátiles (es decir, laptop), tabletas, módulos de conexión, teléfonos móviles tal como los denominados teléfonos "inteligentes", las denominadas smartpads, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales,

consolas de videojuegos, dispositivos de transmisión en tiempo real de videos o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden equiparse para comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de video codificados para decodificarlos a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo fuente 12 transmita datos de video codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de video codificados pueden modularse de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o 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 con base en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

Alternativamente, los datos codificados pueden enviarse desde la interfaz de salida 22 a un dispositivo de almacenamiento 26. De manera similar, puede accederse a los datos codificados desde el dispositivo de almacenamiento 26 mediante la interfaz de entrada. El dispositivo de almacenamiento 26 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tal 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 video codificados. En un ejemplo adicional, el dispositivo de almacenamiento 26 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede contener el video codificado generado por el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a los datos de video almacenados desde el dispositivo de almacenamiento 26 a través de transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 14. Los servidores de archivos ilustrativos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de video codificados a través de cualquier conexión de datos estándar, incluida 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 ambos que sea adecuada para acceder a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el dispositivo de almacenamiento 26 puede ser una transmisión en tiempo real, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de video en soporte de cualquiera de una variedad de aplicaciones multimedia, tal como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video en tiempo real, por ejemplo, a través de Internet, codificación de video digital para almacenamiento en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para soportar transmisión de video unidireccional o bidireccional para admitir aplicaciones tal como transmisión en tiempo real de video, reproducción de video, transmisión de video y/o telefonía de video.

En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye una fuente de video 18, un codificador de video 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo fuente 12, la fuente de video 18 puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contiene video capturado previamente, una interfaz de alimentación de video para recibir video de un proveedor de contenido de video, y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como fuente de video, o una combinación de tales fuentes. Como un ejemplo, si la fuente de video 18 es una cámara de video, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de video en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

El video capturado, precapturado o generado por ordenador puede ser codificado por el codificador de video 20. Los datos de video codificados pueden transmitirse directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo fuente 12. Los datos de video codificados también pueden (o alternativamente) almacenarse en el dispositivo de almacenamiento 26 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para decodificar y/o reproducir.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de video 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de video codificados a través del enlace 16. Los datos de video codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 26,

pueden incluir una variedad de elementos de sintaxis generados por el codificador de video 20 para su uso por un decodificador de video, tal como el decodificador de video 30, al decodificar los datos de video. Tales elementos de sintaxis pueden incluirse con los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

El dispositivo de visualización 32 puede estar integrado o ser externo al dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interactuar con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de video decodificados a un usuario y puede comprender cualquiera de una variedad de dispositivos de visualización tal como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánica (OLED), u otro tipo de dispositivo de visualización.

El codificador de video 20 y el decodificador de video 30 pueden funcionar de acuerdo con un estándar de compresión de video, tal como el estándar recientemente finalizado Codificación de Video de Alta Eficiencia (HEVC), y pueden ajustarse al Modelo de Prueba HEVC (HM). Alternativamente, el codificador de video 20 y el decodificador de video 30 pueden operar de acuerdo con otros estándares de propiedad o de la industria, tal como el estándar ITU-T H.264, también conocido como MPEG-4, Parte 10, Codificación de Video Avanzada (AVC) o extensiones de tales estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular. Otros ejemplos de estándares de compresión de video incluyen MPEG-2 e ITU-T H.263.

Las técnicas de esta divulgación pueden usar terminología HEVC para facilitar la explicación. Sin embargo, no debe asumirse que las técnicas de esta divulgación están limitadas a HEVC y, de hecho, se contempla explícitamente que las técnicas de esta divulgación pueden implementarse en estándares sucesores de HEVC y sus extensiones.

Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de video 20 y el decodificador de video 30 pueden integrarse cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y video en tanto una transmisión de datos común como en transmisiones de datos separados. Si corresponde, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo multiplexor ITU H.223 u otros protocolos tal como el protocolo de datagramas de usuario (UDP).

El codificador de video 20 y el decodificador de video 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos de codificador adecuados, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matriz de puertas lógicas programable en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquiera de sus combinaciones. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador adecuado y no transitorio y ejecutar las instrucciones en hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de video 20 y decodificador de video 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.

Como se presentó anteriormente, el JCT-VC ha finalizado recientemente el desarrollo del estándar HEVC. Los esfuerzos de estandarización de HEVC se basaron en un modelo en evolución de un dispositivo de codificación de video denominado Modelo de Prueba HEVC (HM). El HM presupone varias capacidades adicionales de los dispositivos de codificación de video con relación a los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y cinco modos de codificación de intrapredicción.

En HEVC y otras especificaciones de codificación de video, una secuencia de video típicamente incluye una serie de imágenes. Las imágenes también pueden denominarse "fotogramas". Una imagen puede incluir tres arreglos de muestra, denominadas  $S_L$ ,  $S_{Cb}$  y  $S_{Cr}$ .  $S_L$  es un arreglo bidimensional (es decir, un bloque) de muestras de luminancia.  $S_{Cb}$  es un arreglo bidimensional de muestras de crominancia Cb.  $S_{Cr}$  es un arreglo bidimensional de muestras de crominancia Cr. Las muestras de crominancia también pueden denominarse en la presente memoria muestras de "croma". En otros casos, una imagen puede ser monocromática y solo puede incluir una serie de muestras de luminancia.

Para generar una representación codificada de una imagen, el codificador de video 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque de árbol de codificación de muestras de luminancia, dos bloques de árbol de codificación correspondientes de muestras cromáticas y estructuras de sintaxis usadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un único bloque de árbol de codificación y estructuras de sintaxis usadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras  $N \times N$ . Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación mayor" (LCU). Las CTU de HEVC pueden ser ampliamente análogas a los macrobloques de otros estándares, tal como H.264/AVC. Sin embargo, una CTU no

está necesariamente limitada a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un segmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de escaneo ráster.

Para generar una CTU codificada, el codificador de video 20 puede realizar recursivamente una partición de cuatro árboles en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación puede ser un bloque de muestras NxN. Una CU puede comprender un bloque de codificación de muestras de luminancia y dos bloques de codificación correspondientes de muestras cromáticas de una imagen que tiene un arreglo de muestras de luminancia, un arreglo de muestras de Cb y un arreglo de muestras de Cr, y estructuras de sintaxis usadas para codificar las muestras de bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un solo bloque de codificación y estructuras de sintaxis usadas para codificar las muestras del bloque de codificación.

El codificador de video 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque de muestras rectangular (es decir, cuadrado o no cuadrado) en el que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luminancia, dos bloques de predicción correspondientes de muestras cromáticas 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 solo bloque de predicción y estructuras de sintaxis usadas para predecir el bloque de predicción. El codificador de video 20 puede generar bloques de predicción de luminancia, Cb y Cr para los bloques de predicción de luminancia, Cb y Cr de cada PU de la CU.

El codificador de video 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de video 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas de la imagen asociada con la PU. Si el codificador de video 20 usa la interpredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas de una o más imágenes distintas de la imagen asociada con la PU.

Después de que el codificador de video 20 genera bloques de predicción de luminancia, Cb y Cr para una o más PU de una CU, el codificador de video 20 puede generar un bloque residual de luminancia para la CU. Cada muestra en el bloque residual de luminancia de la CU indica una diferencia entre una muestra de luminancia en uno de los bloques de luminancia predictivos de la CU y una muestra correspondiente en el bloque de codificación de luminancia original de la CU. Además, el codificador de video 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques de Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de video 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques de Cr predictivos de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

Además, el codificador de video 20 puede usar la partición de cuatro árboles para descomponer los bloques residuales de luminancia, Cb y Cr de una CU en uno o más bloques de transformación de luminancia, Cb y Cr. Un bloque de transformación es un bloque de muestras rectangular (por ejemplo, cuadrado o no cuadrado) en el que se aplica la misma transformación. Una unidad de transformación (TU) de una CU puede comprender un bloque de transformación de muestras de luminancia, dos bloques de transformación correspondientes de muestras cromáticas y estructuras de sintaxis usadas para transformar las muestras de bloques de transformación. Por lo tanto, cada TU de una CU puede asociarse con un bloque de transformación de luminancia, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luminancia asociado con la TU puede ser un subbloque del bloque residual de luminancia de la CU. El bloque de transformación de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformación de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un solo bloque de transformación y estructuras de sintaxis usadas para transformar las muestras del bloque de transformación.

El codificador de video 20 puede aplicar una o más transformaciones a un bloque de transformación de luminancia de una TU para generar un bloque de coeficiente de luminancia para la TU. Un bloque de coeficientes puede ser un arreglo bidimensional de coeficientes de transformación. Un coeficiente de transformación puede ser una cantidad escalar. El codificador de video 20 puede aplicar una o más transformaciones a un bloque de transformación de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de video 20 puede aplicar una o más transformaciones a un bloque de transformación de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luminancia, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de video 20 puede cuantificar el bloque de coeficientes. La cuantificación generalmente se refiere a un proceso en el que los coeficientes de transformación se

cuantifican para posiblemente reducir la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. Después de que el codificador de video 20 cuantifique un bloque de coeficientes, el codificador de video 20 puede codificar la entropía de los elementos de sintaxis que indican los coeficientes de transformación cuantificados. Por ejemplo, el codificador de video 20 puede realizar una Codificación Aritmética Binaria Adaptada al Contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformación cuantificados.

El codificador de video 20 puede generar un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades NAL. Una unidad NAL es una estructura de sintaxis que contiene una indicación del tipo de datos en la unidad NAL y bytes que contienen esos datos en la forma de RBSP intercalados según sea necesario con bits de prevención de emulación. Cada una de las unidades NAL incluye un encabezado de unidad NAL y encapsula un RBSP. El encabezado de la unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por el encabezado de unidad NAL de una unidad NAL indica el tipo de unidad NAL. Un RBSP puede ser una estructura de sintaxis que contiene un número entero de bytes que está encapsulado dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

Los diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un PPS, un segundo tipo de unidad NAL puede encapsular una RBSP para un segmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para mensajes SEI, y así sucesivamente. Las unidades NAL que encapsulan las RBSP para datos de codificación de video (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse unidades VCL NAL.

El decodificador de video 30 puede recibir un flujo de bits generado por el codificador de video 20. Además, el decodificador de video 30 puede analizar el flujo de bits para obtener elementos de sintaxis del flujo de bits. El decodificador de video 30 puede reconstruir las imágenes de los datos de video en base a, al menos en parte, los elementos de sintaxis obtenidos del flujo de bits. El proceso para reconstruir los datos de video puede ser generalmente recíproco al proceso realizado por el codificador de video 20. Además, el decodificador de video 30 puede cuantificar de forma inversa los bloques de coeficientes asociados con las TU de una CU actual. El decodificador de video 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir bloques de transformación asociados con las TU de la CU actual. El decodificador de video 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformación de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el decodificador de video 30 puede reconstruir la imagen.

En el campo de la codificación de video, es común aplicar filtros para mejorar la calidad de una señal de video decodificada. El filtro puede aplicarse como un posfiltro, donde el fotograma filtrado no se usa para la predicción de fotogramas futuros, o como un filtro en bucle, donde el fotograma filtrado puede usarse para predecir fotogramas futuros. Puede diseñarse un filtro, por ejemplo, minimizando el error entre la señal original y la señal filtrada decodificada. De manera similar a los coeficientes de transformación, los coeficientes del filtro  $h(k,l)$ ,  $k = -K, \dots, K$ ,  $l = -K, \dots, K$  pueden cuantificarse

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

codificarse y enviarse al decodificador. Los *normFactor* suelen ser igual a  $2^n$ . Cuanto mayor sea el valor del *normFactor* cuanto más precisa es la cuantificación, que normalmente da como resultado los coeficientes de filtro cuantificados  $F(k,l)$  proporcionando un mejor rendimiento. Por otro lado, valores mayores del *normFactor* típicamente producen coeficientes  $F(k,l)$  que requieren más bits para transmitir.

En el decodificador, los coeficientes de filtro decodificados  $F(k,l)$  se aplican a la imagen reconstruida  $R(i,j)$  como sigue

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

donde  $i$  y  $j$  son las coordenadas de los píxeles dentro del fotograma.

El ALF se propuso en HEVC y se incluyó en varios borradores de trabajo y software de modelo de prueba, es decir, el modelo de prueba HEVC (o "HM"), aunque el ALF no está incluido en la versión final de HEVC. Entre las tecnologías relacionadas, el diseño del filtro en el modelo de prueba HEVC versión HM-3.0 se consideró generalmente como el diseño más eficiente.

El ALF en HM-3.0 se basa en la optimización del nivel de imagen. Es decir, los coeficientes de filtro se obtienen después de codificar un fotograma completo. Había dos modos para el componente de luminancia, adaptación con base en bloques (BA) y adaptación con base en regiones (RA). Estos dos modos comparten las mismas formas de



filtro y operaciones de filtrado, así como también los elementos de sintaxis. La única diferencia entre ellos es el procedimiento de clasificación.

El filtrado de Bucle Adaptativo (ALF) con adaptación con base en bloques en HM-3.0 es un ejemplo de un filtro en bucle. En el ALF para el componente de luminancia, los bloques 4x4 en la imagen completa se clasifican con base en una direccionalidad (hasta 3 direcciones) del bloque y una actividad laplaciana 2D (hasta 5 valores de actividad) del bloque bajo el supuesto de que debería haber diferentes filtros para las regiones de la imagen que tienen diferentes características. El cálculo de la direccionalidad  $D$  y de la actividad no cuantificada  $A$  se muestra en las ecuaciones (2) a (5):

$$g_v = \sum_{i=0}^3 \sum_{j=0}^3 V_{i,j}, \quad V_{i,j} = |2R(i,j) - R(i,j-1) - R(i,j+1)|, \quad (2)$$

$$g_h = \sum_{i=0}^3 \sum_{j=0}^3 H_{i,j}, \quad H_{i,j} = |2R(i,j) - R(i-1,j) - R(i+1,j)|, \quad (3)$$

$$D = \begin{cases} 1 & g_h > 2 * g_v \\ 2 & g_v > 2 * g_h \\ 0 & \text{de cualquier otra manera} \end{cases} \quad (4)$$

$$A = \sum_{i=0}^3 \sum_{j=0}^3 \left( \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} (V_{k,l} + H_{k,l}) \right), \quad (5)$$

donde  $R(i,j)$  indica un píxel reconstruido con coordenadas relativas  $(i,j)$  en la parte superior izquierda de un bloque 4x4.  $A$  se cuantifica además en el intervalo de 0 a 4 inclusive, tal como se describió en HM-3.0, y el valor cuantificado de  $A$  será denotado como  $\hat{A}$ .

En total, cada bloque puede clasificarse en una de 15 (5x3) clases y se asigna un índice a cada bloque 4x4 de acuerdo con el valor de  $D$  y  $\hat{A}$  del bloque:  $5D + \hat{A}$ . Por lo tanto, podrían señalarse hasta 15 conjuntos de parámetros ALF para el componente de luminancia de una imagen. Para ahorrar el costo de señalización, las clases pueden fusionarse a lo largo del valor del índice de clase. Para cada clase (o clase fusionada), puede señalarse un conjunto de coeficientes de filtro. Adicionalmente, en HM-3.0, se señala una bandera en un nivel de CU para indicar si se aplica el ALF a la CU.

De acuerdo con las técnicas de esta divulgación, un dispositivo de codificación de video (tal como un codificador de video 20 o un decodificador de video 30) puede realizar el filtrado de datos de video decodificados. Tenga en cuenta que la unidad para señalar los coeficientes del filtro puede ser una imagen/fotograma, un segmento, un mosaico u otros. Para simplificar, las descripciones siguientes tratan principalmente la unidad como una imagen. Sin embargo, las técnicas de esta divulgación no se limitan a la señalización ALF a nivel de imagen. Como se analiza con mayor detalle a continuación, después de que el codificador de video 20 codifica una imagen de datos de video, el codificador de video 20 puede decodificar subsecuentemente la imagen, por ejemplo, para su uso como imagen de referencia. De esta manera, el codificador de video 20 codifica y decodifica datos de video. El codificador de video 20 y el decodificador de video 30 pueden realizar un filtrado como un proceso de filtrado "en bucle", aplicando el ALF a imágenes decodificadas que se usan subsecuentemente como imágenes de referencia para imágenes codificadas y/o decodificadas subsecuentemente. Las descripciones en esta divulgación de un "codificador de video" deben entenderse como descripciones de un codificador de video, tal como el codificador de video 20, y un decodificador de video, tal como el decodificador de video 30.

De acuerdo con las técnicas de esta divulgación, un codificador de video (tal como un codificador de video 20 y/o un decodificador de video 30) puede construir filtros para varias clases de bloques. El codificador de video puede clasificar un bloque en base a, por ejemplo, la direccionalidad de un gradiente calculado a partir de los píxeles del bloque. El codificador de video puede calcular uno o más de un gradiente de dirección horizontal, un gradiente de dirección vertical, un gradiente de dirección diagonal de 45 grados o un gradiente de dirección diagonal de 135 grados. En general, el codificador de video puede construir un filtro diferente para cada clase de bloque. Sin embargo, en algunos casos, el codificador de video puede calcular un conjunto de coeficientes de filtro usados por dos o más clases de bloques. Por ejemplo, el codificador de video puede calcular una pluralidad de conjuntos de coeficientes de filtro. El codificador de video puede codificar además un índice para cada una de las clases de bloques (o un subconjunto de las clases de bloques) identificando un conjunto de coeficientes de filtro en la

pluralidad de conjuntos de coeficientes de filtro. Por lo tanto, en el caso donde el codificador de video codifique el mismo índice para dos o más clases de bloques, esas clases de bloques usarán el mismo conjunto de coeficientes de filtro, es decir, el conjunto de coeficientes de filtro correspondiente al valor del índice codificado.

Adicional o alternativamente, el codificador de video puede indicar que, para al menos una clase de bloques, el filtro se generará mediante el uso del mismo filtro, pero con una transformación geométrica aplicada a una región de soporte del filtro o los coeficientes de filtro. Por lo tanto, incluso para bloques con el mismo índice de clase, el soporte del filtro puede ser diferente debido a transformaciones geométricas. Funcionalmente, aplicar la transformación geométrica a la región de soporte del filtro produce los mismos resultados matemáticos que aplicar la transformación geométrica a los coeficientes del filtro y, por lo tanto, el codificador de video puede aplicar la transformación geométrica a la región de soporte del filtro o los coeficientes del filtro. En este ejemplo, el codificador de video puede codificar, para una clase actual de bloque, datos que identifican un tipo de transformación geométrica (por ejemplo, una rotación, un volteo en diagonal o un volteo en vertical, o una transformación no geométrica), así como también un filtro construido previamente cuyos coeficientes de filtro se usarán para un filtro generado para la clase actual de bloque.

Adicional o alternativamente, el codificador de video puede codificar un valor (por ejemplo, de una bandera) que indica si se usa al menos un filtro fijo para predecir coeficientes de filtro para una clase de bloque. Es decir, la bandera puede indicar, por ejemplo, que los coeficientes de filtro para un filtro de una clase de bloque se predicen a partir de un filtro fijo o, en cambio, se predicen (o heredan) de filtros de una imagen codificada previamente o sin predicción de tanto un filtro fijo como filtros de imágenes previamente codificadas. En algunos ejemplos, el codificador de video puede configurarse con una pluralidad de filtros fijos para cada clase de bloque. En tales ejemplos, el codificador de video puede codificar adicionalmente un índice en la pluralidad de filtros fijos, identificando uno de la pluralidad de filtros fijos que se usará para predecir los coeficientes de filtro de un filtro actual para la clase de bloque, cuando el valor indica que los coeficientes de filtro se predicen a partir de un filtro fijo.

En algunos ejemplos, el codificador de video puede codificar información adicional que indique que ninguno de los coeficientes de filtro para ninguna de las clases se predice a partir de ningún filtro fijo, o que todos los coeficientes de filtro se pronostican a partir de filtros fijos, o que hay algunos coeficientes de filtro predichos de filtros fijos, mientras que otros coeficientes de filtro no se predicen a partir de filtros fijos. Cuando ninguno de los coeficientes de filtro se predice a partir de un filtro fijo, el codificador de video puede evitar (por ejemplo, omitir o prevenir) la codificación de datos que representan un índice en una pluralidad de filtros fijos. Por otro lado, cuando algunos o todos los coeficientes de filtro se predicen a partir de un filtro fijo, el código de video puede codificar además una bandera para cada clase, ya sea que el filtro para una clase se prediga a partir de un filtro fijo o no. Además, el codificador de video puede codificar índices en pluralidades respectivas de filtros fijos para aquellos conjuntos de coeficientes de filtro que se predicen a partir de un filtro fijo, donde los índices identifican el filtro fijo de referencia a partir del cual se van a predecir los coeficientes de filtro correspondientes. Además, el codificador de video puede usar los filtros fijos identificados por los índices para predecir los coeficientes de filtro correspondientes. Después de la predicción, el codificador de video puede codificar además valores residuales (por ejemplo, valores de desplazamiento), que representan diferencias entre los coeficientes de filtro predichos y los valores de coeficiente de filtro reales.

El codificador de video puede aplicar estas técnicas solas o en combinación. Por ejemplo, un codificador de video puede codificar explícitamente una pluralidad de conjuntos de coeficientes de filtro, codificar índices para un primer conjunto de clases de bloques en la pluralidad de conjuntos de coeficientes de filtro y codificar de forma predictiva coeficientes de filtro para filtros de un segundo conjunto de clases de bloques de filtros fijos o filtros de una o más imágenes de referencia. Alternativamente, un codificador de video puede codificar cada uno de una pluralidad de conjuntos de coeficientes de filtro predichos de filtros fijos o de un filtro de una o más imágenes codificadas previamente, y luego para cada una de las clases de bloques, codificar un índice que identifica uno de los conjuntos de coeficientes de filtro. Como otro ejemplo más, un codificador de video puede codificar coeficientes de filtrado para un primer conjunto de clases de bloques mediante el uso de cualquiera de las técnicas anteriores, y codificar información de transformación geométrica para un segundo conjunto de clases de bloques, incluyendo datos que identifican una transformación geométrica y datos que identifican uno de los primeros conjuntos de clases de bloques del que heredar los coeficientes de filtro.

La Figura 2 es un diagrama conceptual que ilustra los 15 grupos (también llamados clases) usados para la clasificación BA (adaptación con base en bloques). En el ejemplo de la Figura 2, los filtros se asignan a intervalos de valores para una métrica de actividad (es decir, Rango 0 a Rango 4) y una métrica de dirección. La métrica de dirección en la Figura 2 se muestra con valores de Sin dirección, Horizontal y Vertical, que pueden corresponder a los valores de 0, 1 y 2 anteriores de la ecuación 3. El ejemplo particular de la Figura 2 muestra seis filtros diferentes (es decir, Filtro 1, Filtro 2... Filtro 6) como asignados a los 15 grupos, pero pueden usarse más o menos filtros de manera similar. Aunque la Figura 2 muestra un ejemplo, con 15 grupos, identificados como grupos 221 a 235, también pueden usarse más o menos grupos. Por ejemplo, en lugar de cinco intervalos para la métrica de actividad, pueden usarse más o menos intervalos dando como resultado más o menos grupos. Adicional o alternativamente, en lugar de solo tres direcciones, también pueden usarse direcciones adicionales (por ejemplo, una dirección de 45 grados y una dirección de 135 grados). Si, por ejemplo, se usaran 5 intervalos de la métrica de actividad junto con

cinco direcciones (por ejemplo, sin dirección, horizontal, vertical, 45 diagonales y 135 diagonales), entonces se usarían 25 grupos para la clasificación, con cada uno de los 25 grupos se asignan a un filtro.

Para ambos componentes cromáticos de una imagen, puede aplicarse un único conjunto de coeficientes de filtro. Los coeficientes de filtro de las imágenes de referencia pueden almacenarse y permitir que se reutilicen como coeficientes de filtro de una imagen actual. La imagen actual puede optar por usar coeficientes de filtro almacenados para las imágenes de referencia y omitir la señalización de coeficientes de filtro. En este caso, sólo es necesario señalar un índice de una de las imágenes de referencia, y los coeficientes de filtro almacenados de la imagen de referencia indicada se heredan para la imagen actual.

De acuerdo con las técnicas de esta divulgación, antes de filtrar, un codificador de video (por ejemplo, el codificador de video 20 o el decodificador de video 30) puede aplicar ciertas transformaciones geométricas, tal como rotación, diagonal y volteo en vertical, a los píxeles en la región de soporte del filtro (píxeles que se multiplican por los coeficientes filtrados en la Ec. (1)) en función de la orientación del gradiente de los píxeles reconstruidos antes de realizar el ALF. Estas transformaciones pueden aumentar la similitud entre diferentes regiones dentro de la imagen, por ejemplo, su direccionalidad. Esto puede reducir el número de filtros que deben enviarse al decodificador, reduciendo así el número de bits necesarios para representarlos o, alternativamente, reduciendo el error de reconstrucción. Aplicar las transformaciones a las regiones de soporte de filtros equivale a aplicar las transformaciones directamente a los coeficientes de filtro. Por lo tanto, el codificador de video puede, en cambio, aplicar las transformaciones a los coeficientes del filtro en lugar de a las regiones de soporte del filtro.

De acuerdo con otra técnica de esta divulgación, que puede usarse conjuntamente con o independientemente de otras técnicas ya introducidas, para reducir el número de bits requeridos para representar los coeficientes de filtro, un codificador de video puede fusionar diferentes clases de bloques. Sin embargo, a diferencia de HM-3.0, cualquier conjunto de clases puede fusionarse, incluso las clases que tienen valores no consecutivos de  $C$ . La información sobre qué clases se fusionan se proporciona enviando, para cada una de las clases (por ejemplo, cada una de las 25 clases) un índice  $i_c$ . Las clases que tienen el mismo índice  $i_c$  comparten el mismo filtro  $f(k,l)$ . Por lo tanto, un codificador de video puede generar una pluralidad de conjuntos de coeficientes de filtro, y para cada una de una pluralidad de clases, datos de código que representan un valor de índice  $i_c$ , donde  $C$  puede ser un valor entre 1 y el número de clases, e  $i_c$  es un índice en los conjuntos generados de coeficientes de filtro que identifica el conjunto de coeficientes de filtro para la clase  $C$ .

De acuerdo con otra técnica de esta divulgación, que puede usarse de forma conjunta o independiente de otras técnicas ya introducidas, otra modificación relativa al ALF como se describió en HM-3.0 es que un codificador de video (por ejemplo, codificador de video 20 o decodificador de video 30) puede usar un conjunto de  $n$  filtros fijos, obtenidos, por ejemplo, mediante entrenamiento fuera de línea, asignados a cada clase para codificar de forma predictiva los coeficientes de filtro. Para cada clase para la cual se predicen coeficientes de filtro a partir de un filtro fijo, el codificador de video puede codificar datos que representan cuál de los  $n$  filtros fijos se usa. En otro ejemplo, algunas clases pueden tener el mismo conjunto de  $n$  filtros fijos asignados o incluso el mismo conjunto de  $n$  filtros fijos puede asignarse a todas las clases. Incluso cuando se selecciona el filtro fijo para una clase determinada, los coeficientes del filtro adaptativo todavía pueden señalarse para esta clase. Los coeficientes del filtro que pueden aplicarse a la imagen reconstruida son, en este caso, la suma de los coeficientes señalizados  $f(k,l)$  y los coeficientes de filtro fijos. El número de clases que pueden compartir los mismos coeficientes señalados  $f(k,l)$  incluso si se eligieron diferentes filtros fijos para ellos.

Ahora se describirán ejemplos de implementaciones de las técnicas introducidas anteriormente. En un ejemplo, cada bloque  $N \times N$  se clasifica en una de las 25 clases con base en su direccionalidad  $D$  y su valor cuantificado de la actividad  $\hat{A}$ :

$$C = 5D + \hat{A}. \quad (6)$$

Los valores del gradiente horizontal, el gradiente vertical y los dos gradientes diagonales se calculan mediante el uso del Laplaciano 1-D:

$$g_v = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} V_{k,l}, \text{ donde} \quad (7)$$

$$V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|,$$

$$g_h = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} H_{k,l}, \text{ donde} \quad (8)$$

$$H_{k,l} = |2R(k, l) - R(k-1, l) - R(k+1, l)|,$$

$$g_{d1} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D1_{k,l}, \text{ donde} \quad (9)$$

$$D1_{k,l} = |2R(k, l) - R(k-1, l-1) - R(k+1, l+1)|,$$

$$g_{d2} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D2_{k,l}, \text{ donde} \quad (10)$$

$$D2_{k,l} = |2R(k, l) - R(k-1, l+1) - R(k+1, l-1)|.$$

En estos ejemplos,  $g_v$  representa un gradiente vertical,  $g_h$  representa un gradiente horizontal,  $g_{d1}$  representa un gradiente diagonal de 45 grados y  $g_{d2}$  representa un gradiente diagonal de 135 grados.

Los índices  $i$  y  $j$  se refieren a las coordenadas del píxel superior izquierdo en el bloque  $N \times N$ . Para asignar la direccionalidad  $D$  de la relación de máximo y mínimo de los gradientes horizontal y vertical

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v), \quad (11)$$

y la relación de máximo y mínimo de dos gradientes diagonales

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}), \quad (12)$$

se comparan entre sí y con un conjunto de umbrales  $t_1$  y  $t_2$ :

Etapla 1. Si ambos  $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$  y  $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ ,

$D$  se establece en 0 (el bloque se clasifica como "textura").

Etapla 2. Si  $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$

se continúa desde la Etapla 3; de cualquier otra manera, se continua desde la Etapla 4.

Etapla 3. Si  $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ ,

$D$  se establece en 2, de cualquier otra manera  $D$  se establece en 1 (el bloque se clasifica como "fuerte horizontal/vertical" o "horizontal/vertical", respectivamente).

Etapla 4. Si  $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ ,

$D$  se establece en 4, de cualquier otra manera  $D$  se establece en 3 (el bloque se clasifica como "diagonal fuerte" o "diagonal", respectivamente).

El valor de actividad  $A$  se calcula como:

$$A = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} (V_{k,l} + H_{k,l}). \quad (13)$$

$A$  se cuantifica además en el intervalo de 0 a 4 inclusive, y el valor cuantificado se denota como  $\hat{A}$ :

$$A = \min(15, (24 \cdot A) \gg 13) \text{ and } \hat{A} = Q[A], \text{ donde} \quad (14)$$

$$Q = \{0, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4\}.$$

Pueden señalizarse hasta 25 filtros, uno para cada clase. Para reducir el número de bits necesarios para representar los coeficientes de filtro, pueden fusionarse diferentes clases. Sin embargo, a diferencia de HM-3.0, cualquier conjunto de clases puede fusionarse, incluso las clases que tienen valores no consecutivos de  $C$ . La información sobre qué clases se fusionan se proporciona codificando, para cada una de las 25 clases, un índice  $i_c$ . Las clases que tienen el mismo índice  $i_c$  comparten el mismo filtro  $f(k, l)$ .

Antes de filtrar cada bloque, pueden aplicarse transformaciones geométricas simples tal como la rotación o el volteo en diagonal y vertical a los coeficientes del filtro  $f(k, l)$  en función de los valores de gradiente calculados para ese bloque. Esto equivale a aplicar estas transformaciones a los píxeles en la región de soporte del filtro. La idea es hacer más similares los diferentes bloques a los que se aplica el ALF alineando su direccionalidad. Las transformaciones geométricas se definen como

$$\begin{aligned} \text{Vuelco en Diagonal: } f_D(k, l) &= f(l, k), \\ \text{Vuelco en Vertical: } f_V(k, l) &= f(k, K - l - 1), \\ \text{Rotación Derecha: } f_R(k, l) &= f(K - l - 1, k), \end{aligned} \quad (15)$$

donde  $K$  es el tamaño del filtro y  $0 \leq k, l \leq K - 1$  son las coordenadas de los coeficientes, de manera que la ubicación  $(0, 0)$  está en la esquina superior izquierda y la ubicación  $(K - 1, K - 1)$  está en la esquina inferior derecha. El mapeo entre los valores de gradiente calculados mediante el uso de las Ecs. (7)-(10) y las transformaciones geométricas especificadas en la Ec. (15) se muestra en la Tabla 1.

Tabla 1: Mapeo del gradiente calculado para un bloque  $N \times N$  y el filtro

Valores de gradiente	Coeficientes de filtro
$g_{d2} < g_{d1}$ y $g_h < g_v$	$f(k, l)$
$g_{d2} < g_{d1}$ y $g_v < g_h$	$f_D(k, l)$
$g_{d1} < g_{d2}$ y $g_h < g_v$	$f_V(k, l)$
$g_{d1} < g_{d2}$ y $g_v < g_h$	$f_R(k, l)$

En algunos ejemplos, la predicción de coeficientes de filtro puede ser de filtros fijos. La predicción temporal de los coeficientes de filtro puede mejorar la eficacia de la codificación para fotogramas intercodificados. Para mejorar la eficiencia de la codificación cuando la predicción temporal no está disponible (intrafotogramas), se asigna un conjunto de  $n$  filtros fijos a cada clase. En algunos ejemplos, para cada una de las 25 clases, se señala una bandera si se usa el filtro fijo y, si es necesario, un índice del filtro elegido. En algunos ejemplos, el valor de la bandera puede indicar si se usa un filtro fijo. En algunos ejemplos, si es necesario, puede señalizarse un índice que representa el filtro elegido. Alternativamente, puede usarse uno de los siguientes tres casos definidos ilustrativo. En un primer caso, ninguno de los filtros de las 25 clases se predice a partir del filtro fijo. En un segundo caso, todos los filtros de las clases se predicen a partir del filtro fijo. En un tercer caso, los filtros asociados con algunas clases se predicen a partir de filtros fijos y los filtros asociados con el resto de las clases no se predicen a partir de los filtros fijos.

Puede codificarse una variable para indicar cuál de los tres casos ilustrativo anteriores se aplica. Además, puede aplicarse lo siguiente:

- Si la variable indica el caso 1, no hay necesidad de señalizar más el índice del filtro fijo.
- De cualquier otra manera, si la variable indica el caso 2, se codifica un índice del filtro fijo seleccionado para cada clase.
- De cualquier otra manera, si la variable indica el caso 3, puede señalizarse primero un bit para cada clase, y si se usa el filtro fijo, puede señalizarse además el índice.

Incluso cuando se selecciona el filtro fijo para una clase determinada, los coeficientes del filtro adaptativo todavía pueden enviarse para esta clase. Los coeficientes del filtro que se aplicarán a la imagen reconstruida son en este caso la suma de los coeficientes  $f(k, l)$  señalados y los coeficientes del filtro fijos. El número de clases puede compartir los mismos coeficientes  $f(k, l)$  incluso si se eligieron diferentes filtros fijos para ellos. En una alternativa, los

filtros fijos aplicados a intrafotogramas también pueden aplicarse a fotogramas intercodificados, tal como segmentos P o B.

La Figura 3 es un diagrama conceptual que ilustra un patrón ilustrativo para señalar diferencias de coeficientes de filtro. Las siguientes técnicas pueden usarse para definir uno o más filtros fijos, por ejemplo, valores de coeficientes para los filtros fijos. En un ejemplo, los filtros fijos se derivan del entrenamiento fuera de línea y solo se definen los coeficientes de filtro para la derivación de filtro más grande (por ejemplo, forma de filtro de diamante de 9x9). Antes de codificar las diferencias entre los coeficientes de filtro derivados y los filtros fijos seleccionados, un codificador de video puede codificar primero un índice que indica el patrón de diferencias de coeficientes de filtro de señalización. Por ejemplo, el filtro fijo para una forma de filtro de diamante simétrico de 9x9 se representa en la Figura 3, en el que el coeficiente de filtro está indicado por  $F_i$  ( $i$  está en el intervalo de 0 a 20, inclusive). Cuando el índice del patrón señalado indica una forma de filtro de diamante de 5x5, tal como se muestra en la Figura 3, y que los coeficientes de filtro se predicen a partir de un filtro fijo, los valores de los coeficientes de filtro derivados ubicados en las áreas grises de la Figura 3 se mantienen sin cambios y no es necesario señalar las diferencias para estas posiciones (es decir, las diferencias para estas posiciones grises son iguales a 0). En este caso, el codificador de video codifica los datos que definen las diferencias entre el coeficiente derivado y los coeficientes de filtro fijos ( $F_0 \sim F_6$ ). Alternativamente, el codificador de video no necesita codificar datos que definan la diferencia del coeficiente de DC (es decir,  $F_6$ ), sino que deriva la diferencia del coeficiente de DC en otros valores de diferencia codificados. Tenga en cuenta que, en este caso, el proceso de filtrado de diamantes 9x9 siempre se aplica cuando los coeficientes de filtro se predicen a partir del filtro fijo.

En otro ejemplo, pueden predefinirse M versiones de filtros fijos, en el que M indica el número total de formas de filtro admitidas. El codificador de video puede codificar primero un índice de la forma de filtro seleccionada, y luego codificar datos que representan los índices de filtros fijos seleccionados asociados con la forma de filtro. Tenga en cuenta que, en este caso, se admiten varios procesos de filtrado diferentes (en función del valor de M).

La Figura 4 es un diagrama de bloques que ilustra un codificador de video 20 ilustrativo que puede implementar las técnicas descritas en esta divulgación. El codificador de video 20 puede realizar la intracodificación y la intercodificación de bloques de video dentro de segmentos de video. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el video dentro de un fotograma o imagen de video determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el video dentro de fotogramas o imágenes adyacentes de una secuencia de video. El intramodo (modo I) puede referirse a cualquiera de varios modos de compresión basados en el espacio. Los intermodos, tal como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión basados en el tiempo.

En el ejemplo de la Figura 4, el codificador de video 20 incluye una memoria de datos de video 33, una unidad de partición 35, una unidad de procesamiento de predicción 41, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54, una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye la unidad de estimación del movimiento (MEU) 42, la unidad de compensación del movimiento (MCU) 44 y la unidad de intrapredicción 46. Para la reconstrucción de los bloques de video, el codificador de video 20 incluye además la unidad de cuantificación inversa 58, la unidad de procesamiento de transformación inversa 60, el sumador 62, la unidad del ALF 64 y la memoria intermedia de imágenes decodificadas (DPB) 66.

Como se muestra en la Figura 4, el codificador de video 20 recibe datos de video y almacena los datos de video recibidos en la memoria de datos de video 33. La memoria de datos de video 33 puede almacenar datos de video para codificarlos por los componentes del codificador de video 20. Los datos de video almacenados en la memoria de datos de video 33 pueden obtenerse, por ejemplo, de la fuente de video 18. La DPB 66 puede ser una memoria de imágenes de referencia que almacena datos de video de referencia para su uso en la codificación de datos de video mediante el codificador de video 20, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de video 33 y la DPB 66 pueden formarse por cualquiera de una variedad de dispositivos de memoria, tal como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de video 33 y la DPB 66 pueden proporcionarse por el mismo dispositivo de memoria o por dispositivos de memoria separados. En varios ejemplos, la memoria de datos de video 33 puede estar en un chip con otros componentes del codificador de video 20, o fuera de un chip con relación a esos componentes.

La unidad de particionamiento 35 recupera los datos de video de la memoria de datos de video 33 y divide los datos de video en bloques de video. Esta división también puede incluir la división en segmentos, mosaicos u otras unidades más grandes, así como también la división de bloques de video, por ejemplo, de acuerdo con una estructura de cuatro árboles de LCU y CU. El codificador de video 20 generalmente ilustra los componentes que codifican bloques de video dentro de un segmento de video a codificar. El segmento puede dividirse en múltiples bloques de video (y posiblemente en conjuntos de bloques de video denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno de una pluralidad de modos de codificación posibles, tal como uno de una pluralidad de modos de intracodificación o uno de una pluralidad de modos de codificación mutua,

para el bloque de video actual con base en resultados de error (por ejemplo, tasa de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra o inter codificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

La unidad de intrapredicción 46 dentro de la unidad de procesamiento de predicción 41 puede realizar la intrapredicción del bloque de video actual con relación a uno o más bloques vecinos en el mismo fotograma o segmento que el bloque actual a codificar para proporcionar compresión espacial. La unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44 dentro de la unidad de procesamiento de predicción 41 realizan la interpredicción del bloque de video actual con relación a uno o más bloques de referencia en una o más imágenes de referencia para proporcionar compresión temporal.

La unidad de estimación del movimiento 42 puede configurarse para determinar el modo de interpredicción para un segmento de video de acuerdo con un patrón predeterminado para una secuencia de video. El patrón predeterminado puede designar segmentos de video en la secuencia como segmentos P o segmentos B. La unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44 pueden estar muy integradas, pero se ilustran por separado con fines conceptuales. La estimación del movimiento, realizada por la unidad de estimación del movimiento 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de video. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de video dentro de una imagen o fotograma de video actual con relación a un bloque predictivo dentro de una imagen de referencia.

Un bloque predictivo es un bloque que coincide estrechamente con la PU del bloque de video que se codificará en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otra métrica de diferencia. En algunos ejemplos, el codificador de video 20 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la DPB 66. Por ejemplo, el codificador de video 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionarios de la imagen de referencia. Por lo tanto, la unidad de estimación del movimiento 42 puede realizar una búsqueda de movimiento con relación a las posiciones de píxeles completos y posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios.

La unidad de estimación del movimiento 42 calcula un vector de movimiento para una PU de un bloque de video en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la DPB 66. La unidad de estimación del movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación del movimiento 44.

La compensación del movimiento, realizada por la unidad de compensación del movimiento 44, puede implicar buscar o generar el bloque predictivo con base en el vector de movimiento determinado por la estimación del movimiento, posiblemente realizando interpolaciones con precisión de subpíxeles. Al recibir el vector de movimiento para la PU del bloque de video actual, la unidad de compensación del movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de video 20 forma un bloque de video residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de video actual que se está codificando, formando valores de diferencia de píxeles. Los valores de diferencia de píxeles forman datos residuales para el bloque y pueden incluir componentes de diferencia de luminancia y crominancia. El sumador 50 representa el componente o componentes que realizan esta operación de substracción. La unidad de compensación del movimiento 44 también puede generar elementos de sintaxis asociados con los bloques de video y el segmento de video para su uso por el decodificador de video 30 al decodificar los bloques de video del segmento de video.

Después de que la unidad de procesamiento de predicción 41 genera el bloque predictivo para el bloque de video actual, ya sea a través de intrapredicción o interpredicción, el codificador de video 20 forma un bloque de video residual restando el bloque predictivo del bloque de video actual. Los datos de video residuales en el bloque residual pueden incluirse en una o más TU y aplicarse a la unidad 52 de procesamiento de transformación. La unidad de procesamiento de transformación 52 transforma los datos de video residuales en coeficientes de transformación residual mediante el uso de una transformación, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformación 52 puede convertir los datos de video residuales de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia.

La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede entonces realizar una exploración de la

matriz que incluye los coeficientes de transformación cuantificados. Alternativamente, la unidad de codificación por entropía 56 puede realizar la exploración.

Después de la cuantificación, la entropía de la unidad de codificación por entropía 56 codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto con base en la sintaxis (SBAC), codificación por entropía de partición de intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Después de la codificación por entropía mediante la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse al decodificador de video 30 o archivarse para su posterior transmisión o recuperación mediante el decodificador de video 30. La unidad de codificación por entropía 56 también puede codificar por entropía de los vectores de movimiento y los otros elementos de sintaxis para el segmento de video actual que se codifica.

La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles para su uso posterior como bloque de referencia de una imagen de referencia. La unidad de compensación del movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación del movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación del movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por el movimiento producido por la unidad de compensación del movimiento 44 para producir un bloque reconstruido.

La unidad del ALF 64 filtra el bloque reconstruido (por ejemplo, la salida del sumador 62) y almacena el bloque reconstruido filtrado en la DPB 66 para su uso como bloque de referencia. El bloque de referencia puede usarse por la unidad de estimación del movimiento 42 y por la unidad de compensación del movimiento 44 como un bloque de referencia para predecir un bloque en un fotograma de video o en una imagen subsecuente. Aunque no se muestra explícitamente en la Figura 4, el codificador de video 20 puede incluir filtros adicionales tal como uno o más de un filtro de desbloqueo, un filtro de desplazamiento adaptativo de muestra (SAO) u otro tipo de filtro de bucle. Un filtro de desbloqueo puede, por ejemplo, aplicar un filtrado de desbloqueo para filtrar los límites del bloque para eliminar los artefactos de bloqueo del video reconstruido. Un filtro SAO puede aplicar desplazamientos a los valores de los píxeles reconstruidos para mejorar la calidad general de la codificación. En algunas implementaciones, el SAO puede ser un caso especial o un modo especial de filtrado. También pueden usarse filtros de bucle adicionales (en bucle o en bucle posterior).

La unidad del ALF 64, sola o junto con otros componentes del codificador de video 20, puede configurarse para realizar las diversas técnicas de ALF descritas en esta divulgación. Por ejemplo, la salida del sumador 62 son datos de video decodificados que fueron previamente codificados por otros elementos del codificador de video 20 (tal como la unidad de procesamiento de predicción 41, la unidad de procesamiento de transformación 52 y la unidad de cuantificación 54). Por lo tanto, la unidad del ALF 64 puede filtrar datos de video decodificados de acuerdo con las técnicas de esta divulgación. En particular, la unidad del ALF 64 realiza el filtrado como un proceso de filtrado "en bucle", en el que la unidad del ALF 64 filtra imágenes decodificadas que se usan subsecuentemente como imágenes de referencia, almacenadas en la DPB 66, para usarse por la unidad de procesamiento de predicción 41 para predecir imágenes subsecuentes.

De acuerdo con las técnicas de esta divulgación, la unidad del ALF 64 puede construir filtros para varias clases de bloques. La unidad del ALF 64 puede clasificar un bloque con base en, por ejemplo, la direccionalidad de un gradiente calculado a partir de píxeles del bloque. En general, la unidad del ALF 64 puede construir un filtro diferente para cada clase de bloque. Sin embargo, en algunos casos, la unidad del ALF 64 puede calcular un conjunto de coeficientes de filtro usados por dos o más clases de bloques. Por ejemplo, la unidad del ALF 64 puede calcular una pluralidad de conjuntos de coeficientes de filtro. La unidad del ALF 64 puede codificar además un valor de índice para cada una de las clases de bloques (o un subconjunto de las clases de bloques) identificando un conjunto de coeficientes de filtro en la pluralidad de conjuntos de coeficientes de filtro. La unidad del ALF 64 puede construir una estructura de datos que especifique los índices para cada una de las clases de bloques y proporcionar la estructura de datos a la unidad de codificación por entropía 56 para que sea codificada por entropía e incluida en el flujo de bits. La unidad de codificación por entropía 56 puede codificar por entropía la estructura de datos y añadir la estructura de datos a, por ejemplo, un encabezado de segmento, un conjunto de parámetros de imagen u otra estructura de datos similar. Así, en el caso donde la unidad del ALF 64 codifique el mismo índice para dos o más clases de bloques, esas clases de bloques usarán el mismo conjunto de coeficientes de filtro, es decir, el conjunto de coeficientes de filtro correspondiente al valor de índice codificado.

Adicional o alternativamente, la unidad del ALF 64 puede indicar que, para al menos una clase de bloque, el filtro se generará mediante el uso de los mismos coeficientes de filtro que una clase diferente de bloque, pero con una transformación geométrica aplicada a una región de soporte del filtro o de los coeficientes del filtro. Funcionalmente, aplicar la transformación geométrica a la región de soporte del filtro produce los mismos resultados matemáticos que aplicar la transformación geométrica a los coeficientes del filtro y, por lo tanto, la unidad del ALF 64 puede aplicar la



transformación geométrica a la región de soporte del filtro o los coeficientes del filtro. En este ejemplo, la unidad del ALF 64 puede codificar, para una clase actual de bloque, datos que identifican un tipo de transformación geométrica (por ejemplo, una rotación, un volteo en diagonal o un volteo en vertical), así como también un filtro construido previamente del cual los coeficientes de filtro se usarán para un filtro generado para la clase actual de bloque. De nuevo, la unidad del ALF 64 puede proporcionar tales datos a la unidad de codificación por entropía 56, que puede codificar la entropía de los datos y añadir los datos a una estructura de datos tal como, por ejemplo, un encabezado de segmento o un conjunto de parámetros de imagen.

Adicional o alternativamente, la unidad del ALF 64 puede codificar un valor (por ejemplo, de una bandera) que indica si se usa un filtro fijo para predecir un conjunto de coeficientes de filtro para una clase de bloque. Igualmente, la unidad del ALF 64 puede proporcionar estos valores a la unidad de codificación por entropía 56 para que sean codificados por entropía, por ejemplo, tal como parte de un encabezado de segmento, conjunto de parámetros de imagen u otra estructura de datos similar. La bandera puede indicar, por ejemplo, que los coeficientes de filtro para un filtro de una clase de bloque se predicen a partir de un filtro fijo o, en cambio, se predicen (o heredan) de un filtro de una imagen de referencia. En algunos ejemplos, la unidad del ALF 64 puede configurarse con una pluralidad de filtros fijos para cada clase de bloque. En tales ejemplos, la unidad del ALF 64 puede codificar adicionalmente un índice en la pluralidad de filtros fijos, identificando uno de la pluralidad de filtros fijos que se usará para predecir los coeficientes de filtro de un filtro actual para la clase de bloque, cuando el valor indica que los coeficientes de filtro se predicen a partir de un filtro fijo.

En algunos ejemplos, la unidad del ALF 64 puede codificar más información que indique que ninguno de los coeficientes de filtro para ninguna de las clases se predice a partir de ningún filtro fijo, que todos los coeficientes de filtro se pronostican a partir de un filtro fijo o que hay algunos coeficientes de filtro predichos de un filtro fijo, mientras que otros coeficientes de filtro no se predicen a partir de un filtro fijo. Cuando ninguno de los coeficientes de filtro se predice a partir de un filtro fijo, la unidad del ALF 64 puede evitar (por ejemplo, omitir o prevenir) la codificación de datos que representan un índice en una pluralidad de filtros fijos. Por otro lado, cuando algunos o todos los coeficientes de filtro se predicen a partir de un filtro fijo, la unidad del ALF 64 puede codificar índices en respectivas pluralidades de filtros fijos para esos conjuntos de coeficientes de filtro que se predicen a partir de un filtro fijo, donde los índices identifican el filtro fijo de referencia a partir del cual se van a predecir los coeficientes de filtro correspondientes. Además, la unidad del ALF 64 puede usar los filtros fijos identificados por los índices para predecir los coeficientes de filtro correspondientes. Después de la predicción, la unidad del ALF 64 puede codificar además valores residuales (por ejemplo, valores de desplazamiento), que representan diferencias entre los coeficientes de filtro predichos y los valores de coeficiente de filtro reales.

La unidad del ALF 64 puede aplicar estas técnicas solas o en combinación. Por ejemplo, la unidad del ALF 64 puede codificar explícitamente una pluralidad de conjuntos de coeficientes de filtro, codificar índices para un primer conjunto de clases de bloques en la pluralidad de conjuntos de coeficientes de filtro y codificar de forma predictiva coeficientes de filtro para filtros de un segundo conjunto de clases de bloques de filtros fijos o filtros de una o más imágenes de referencia en la DPB 66. Alternativamente, la unidad del ALF 64 puede codificar cada uno de una pluralidad de conjuntos de coeficientes de filtro predichos de un filtro fijo o de un filtro de una o más imágenes de referencia en la DPB 66, y luego para cada una de las clases de bloques, codificar un índice de identificación de uno de los conjuntos de coeficientes de filtro. Como otro ejemplo más, la unidad del ALF 64 puede codificar coeficientes de filtro para un primer conjunto de clases de bloques mediante el uso de cualquiera de las técnicas anteriores, y codificar la información de la transformación geométrica para un segundo conjunto de clases de bloques, incluidos los datos que identifican una transformación geométrica y los datos que identifican uno de los primeros conjuntos de clases de bloques del que heredar los coeficientes de filtro.

Para determinar cómo seleccionar los índices para clases de bloques, si predecir los coeficientes de filtro a partir de filtros fijos o imágenes de referencia de la DPB 66, y/o si generar un filtro para una clase de bloque mediante el uso de una transformación geométrica como se discutió anteriormente, el codificador de video 20 puede ejecutar una variedad de pases de codificación en una imagen o secuencia de imágenes, y hacer que la unidad del ALF 64 aplique varias técnicas de filtrado a varias clases de bloques en las imágenes. El codificador de video 20 puede calcular métricas de optimización de distorsión de velocidad (RDO) para las imágenes filtradas decodificadas, para determinar qué combinación de coeficientes de filtro y técnicas de predicción de coeficientes de filtro dan como resultado las mejores métricas de RDO. El codificador de video 20 puede seleccionar entonces los coeficientes de filtro y las técnicas de filtrado para cada imagen con base en la combinación de técnicas de filtrado que dan como resultado las mejores métricas de RDO para cada imagen.

La Figura 5 es un diagrama de bloques que ilustra un decodificador de video 30 ilustrativo que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la Figura 5, el decodificador de video 30 incluye la memoria de datos de video 78, la unidad de decodificación por entropía 80, la unidad de procesamiento de predicción 81, la unidad de cuantificación inversa 86, la unidad de procesamiento de transformación inversa 88, el sumador 90 y la memoria intermedia de imágenes decodificadas (DPB) 94. La unidad de procesamiento de predicción 81 incluye la unidad de compensación del movimiento 82 y la unidad de intrapredicción 84. El decodificador de video 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de video 20 de la Figura 4.

Durante el procedimiento de decodificación, el decodificador de video 30 recibe un flujo de bits de video codificado que representa bloques de video de un segmento de video codificado y elementos de sintaxis asociados del codificador de video 20. El decodificador de video 20 almacena el flujo de bits de video codificado recibido en la memoria de datos de video 78. La memoria de datos de video 78 puede almacenar datos de video, tal como un flujo de bits de video codificado, para ser decodificados por los componentes del decodificador de video 30. Los datos de video almacenados en la memoria de datos de video 78 pueden obtenerse, por ejemplo, a través del enlace 16, desde el dispositivo de almacenamiento 26, o desde una fuente de video local, tal como una cámara, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de video 78 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de video codificados de un flujo de bits de video codificado. La DPB 94 puede ser una memoria de imágenes de referencia que almacena datos de video de referencia para su uso en la decodificación de datos de video mediante el decodificador de video 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de video 78 y la DPB 94 pueden formarse por cualquiera de una variedad de dispositivos de memoria, tal como DRAM, SDRAM, MRAM, RRAM u otros tipos de dispositivos de memoria. La memoria de datos de video 78 y la DPB 94 pueden proporcionarse por el mismo dispositivo de memoria o por dispositivos de memoria separados. En varios ejemplos, la memoria de datos de video 78 puede estar en un chip con otros componentes del decodificador de video 30, o fuera de un chip con relación a esos componentes.

La unidad de decodificación por entropía 80 del decodificador de video 30 decodifica por entropía los datos de video almacenados en la memoria de datos de video 78 para generar coeficientes cuantificados, vectores de movimiento y otros elementos de sintaxis. La unidad de decodificación por entropía 80 envía los vectores de movimiento y otros elementos de sintaxis a la unidad de procesamiento de predicción 81. El decodificador de video 30 puede recibir los elementos de sintaxis a nivel de segmento de video y/o a nivel de bloque de video.

Cuando el segmento de video se codifica como un segmento intracodificado (I), la unidad de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de video del segmento de video actual con base en un modo de predicción intraseñalizado y datos de bloques del fotograma o de la imagen actual previamente decodificados. Cuando el fotograma de video se codifica como un segmento intercodificado (por ejemplo, segmento B o segmento P), la unidad de compensación del movimiento 82 de la unidad de procesamiento de predicción 81 produce bloques predictivos para un bloque de video del segmento de video actual con base en los vectores de movimiento y a otros elementos de sintaxis recibidos de la unidad de decodificación por entropía 80. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de video 30 puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, mediante el uso de técnicas de construcción predeterminadas con base en imágenes de referencia almacenadas en la DPB 94.

La unidad de compensación del movimiento 82 determina la información de predicción para un bloque de video del segmento de video actual analizando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de video actual que se está decodificando. Por ejemplo, la unidad de compensación del movimiento 82 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o inter predicción) usado para codificar los bloques de video del segmento de video, un tipo de segmento de interpretación (por ejemplo, segmento B o segmento P), la información de construcción para una o más de las listas de imágenes de referencia para el segmento, los vectores de movimiento para cada bloque de video intercodificado del segmento, el estado de predicción mutua para cada bloque de video intercodificado del segmento, y otra información para decodificar los bloques de video en el segmento de video actual.

La unidad de compensación del movimiento 82 también puede realizar una interpolación con base en filtros de interpolación. La unidad de compensación del movimiento 82 puede usar filtros de interpolación como los usa el codificador de video 20 durante la codificación de los bloques de video para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación del movimiento 82 puede determinar los filtros de interpolación usados por el codificador de video 20 a partir de los elementos de sintaxis recibidos y usar los filtros de interpolación para producir bloques predictivos.

La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de video 20 para cada bloque de video en el segmento de video para determinar un grado de cuantificación e, igualmente, un grado de cuantificación inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformación de entero inverso o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que la unidad de procesamiento de predicción genera el bloque predictivo para el bloque de video actual usando, por ejemplo, intra o inter predicción, el decodificador de video 30 forma un bloque de video reconstruido sumando los bloques residuales de la unidad de procesamiento de transformación inversa 88 con los

correspondientes bloques predictivos generados por unidad de compensación del movimiento 82. El sumador 90 representa el componente o componentes que realizan esta operación de suma. La unidad del ALF 92 filtra el bloque de video reconstruido mediante el uso de, por ejemplo, una o más de las técnicas de filtrado descritas en esta divulgación.

Aunque no se muestra explícitamente en la Figura 5, el decodificador de video 30 también puede incluir uno o más de un filtro de desbloqueo, un filtro SAO u otros tipos de filtros. También pueden usarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o de cualquier otra manera mejorar la calidad del vídeo. Los bloques de video decodificados en un fotograma o imagen dados se almacenan luego en la DPB 94, que almacena imágenes de referencia usadas para el desplazamiento de movimiento subsecuente. La DPB 94 puede ser parte o estar separada de la memoria adicional que almacena el video decodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

La unidad del ALF 92, sola o junto con otros componentes del decodificador de video 30, puede configurarse para realizar las diversas técnicas descritas en esta divulgación, incluidas las técnicas descritas en la sección de reivindicaciones, así como también en otras partes. Por ejemplo, la salida del sumador 90 son datos de video decodificados. Por lo tanto, la unidad del ALF 92 puede filtrar datos de video decodificados de acuerdo con las técnicas de esta divulgación. En particular, la unidad del ALF 92 realiza el filtrado como un proceso de filtrado "en bucle", en el que la unidad del ALF 92 filtra imágenes decodificadas que se usan subsecuentemente como imágenes de referencia, almacenadas en la DPB 94, para usarse por la unidad de procesamiento de predicción 81 para predecir imágenes subsecuentes.

La unidad del ALF 92 puede aplicar las técnicas de esta divulgación solas o en combinación. Por ejemplo, la unidad del ALF 92 puede decodificar explícitamente una pluralidad de conjuntos de coeficientes de filtro, decodificar índices para un primer conjunto de clases de bloques en la pluralidad de conjuntos de coeficientes de filtro y decodificar predictivamente coeficientes de filtro para filtros de un segundo conjunto de clases de bloques de filtros fijos o filtros de una o más imágenes de referencia en la DPB 94, con base en datos decodificados del flujo de bits, tal como un encabezado de segmento o un conjunto de parámetros de imagen. Alternativamente, la unidad del ALF 92 puede decodificar cada uno de una pluralidad de conjuntos de coeficientes de filtro predichos de un filtro fijo o de un filtro de una o más imágenes de referencia en la DPB 94, y luego para cada una de las clases de bloques, decodificar un índice de identificación de uno de los conjuntos de coeficientes de filtro, con base en datos decodificados del flujo de bits, tal como un encabezado de segmento o un conjunto de parámetros de imagen. Como otro ejemplo más, la unidad del ALF 92 puede decodificar coeficientes de filtro para un primer conjunto de clases de bloques mediante el uso de cualquiera de las técnicas anteriores, y decodificar información de transformación geométrica para un segundo conjunto de clases de bloques, incluyendo datos que identifican una transformación geométrica y datos que identifican uno del primer conjunto de clases de bloques del que heredar los coeficientes de filtro, con base en datos decodificados del flujo de bits, tal como un encabezado de segmento o un conjunto de parámetros de imagen. Así, en general,

En algunos ejemplos, la unidad del ALF 92 puede construir filtros para varias clases de bloques. La unidad del ALF 92 puede clasificar un bloque con base en, por ejemplo, la direccionalidad de un gradiente calculado a partir de píxeles del bloque. En general, la unidad del ALF 92 puede construir un ALF diferente para cada clase de bloque. Sin embargo, en algunos casos, la unidad del ALF 92 puede calcular un conjunto de coeficientes de filtro usados por dos o más clases de bloques. Por ejemplo, la unidad del ALF 92 puede calcular una pluralidad de conjuntos de coeficientes de filtro. La unidad del ALF 92 puede decodificar además un valor de índice para cada una de las clases de bloques (o un subconjunto de las clases de bloques) identificando un conjunto de coeficientes de filtro en la pluralidad de conjuntos de coeficientes de filtro. La unidad de decodificación por entropía 80 puede decodificar por entropía una estructura de datos que especifique los índices para cada una de las clases de bloques, y proporcionar la estructura de datos a la unidad del ALF 92. La unidad de decodificación por entropía 56 puede decodificar por entropía la estructura de datos de, por ejemplo, un encabezado de segmento, un conjunto de parámetros de imagen u otra estructura de datos similar. Así, en el caso donde la unidad del ALF 92 decodifica el mismo índice para dos o más clases de bloques, esas clases de bloques usarán el mismo conjunto de coeficientes de filtro, es decir, el conjunto de coeficientes de filtro correspondiente al valor de índice decodificado.

Adicional o alternativamente, la unidad del ALF 92 puede indicar que, para al menos una clase de bloque, el filtro se generará mediante el uso de los mismos coeficientes de filtro que una clase diferente de bloque, pero con una transformación geométrica aplicada a una región de soporte del filtro o de los coeficientes del filtro. Funcionalmente, aplicar la transformación geométrica a la región de soporte del filtro produce los mismos resultados matemáticos que aplicar la transformación geométrica a los coeficientes del filtro y, por lo tanto, la unidad del ALF 92 puede aplicar la transformación geométrica a la región de soporte del filtro o los coeficientes del filtro. En este ejemplo, la unidad del ALF 92 puede decodificar, para una clase actual de bloque, datos que identifican un tipo de transformación geométrica (por ejemplo, una rotación, un volteo en diagonal o un volteo en vertical), así como también un filtro construido previamente del cual los coeficientes de filtro se usarán para un filtro generado para la clase actual de bloque. De nuevo, la unidad de decodificación por entropía 80 puede decodificar por entropía tales datos y

proporcionar los datos decodificados a la unidad del ALF 92. Los datos pueden incluirse en una estructura de datos tal como, por ejemplo, un encabezado de segmento o un conjunto de parámetros de imagen.

Adicional o alternativamente, la unidad de decodificación por entropía 80 puede decodificar por entropía un valor (por ejemplo, de una bandera) que indica si se usa un filtro fijo para predecir un conjunto de coeficientes de filtro para una clase de bloque. Igualmente, la unidad de decodificación por entropía 80 puede proporcionar estos valores a la unidad del ALF 92. La unidad de decodificación por entropía 80 puede decodificar los datos de parte de un encabezado de segmento, un conjunto de parámetros de imagen u otra estructura de datos similar. La bandera puede indicar, por ejemplo, que los coeficientes de filtro para un filtro de una clase de bloque se predicen a partir de un filtro fijo o, en cambio, se predicen (o heredan) de un filtro de una imagen de referencia. En algunos ejemplos, la unidad del ALF 92 puede configurarse con una pluralidad de filtros fijos para cada clase de bloque. En tales ejemplos, la unidad del ALF 92 puede decodificar adicionalmente un índice en la pluralidad de filtros fijos, identificando uno de la pluralidad de filtros fijos que se usará para predecir los coeficientes de filtro de un filtro actual para la clase de bloque, cuando el valor indica que los coeficientes de filtro se predicen a partir de un filtro fijo.

En algunos ejemplos, la unidad del ALF 92 puede recibir además información decodificada por entropía que indica que ninguno de los coeficientes de filtro para ninguna de las clases se predice a partir de ningún filtro fijo, que todos los coeficientes de filtro se predicen a partir de un filtro fijo o que hay algunos coeficientes de filtro predichos a partir de un filtro fijo, mientras que otros coeficientes de filtro no se predicen a partir de un filtro fijo. Cuando ninguno de los coeficientes de filtro se predice a partir de un filtro fijo, la unidad del ALF 92 (o la unidad de decodificación por entropía 80) puede evitar (por ejemplo, omitir o evitar) la decodificación de datos que representan un índice en una pluralidad de filtros fijos. Por otro lado, cuando algunos o todos los coeficientes de filtro se predicen a partir de un filtro fijo, la unidad del ALF 92 puede decodificar índices en respectivas pluralidades de filtros fijos para esos conjuntos de coeficientes de filtro que se predicen a partir de un filtro fijo, donde los índices identifican el filtro fijo de referencia a partir del cual se van a predecir los coeficientes de filtro correspondientes. Además, la unidad del ALF 92 puede usar los filtros fijos identificados por los índices para predecir los coeficientes de filtro correspondientes. Después de la predicción, la unidad del ALF 92 puede decodificar adicionalmente valores residuales (por ejemplo, valores de desplazamiento), que representan diferencias entre los coeficientes de filtro predichos y los valores de coeficiente de filtro reales.

La Figura 6 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 6 se describe con respecto al codificador de video 20 de la Figura 4.

Inicialmente, el codificador de video 20 codifica una imagen (100). Por ejemplo, el codificador de video 20 puede intra o inter predecir los bloques de la imagen, calcular los residuales para los bloques predichos, transformar y cuantificar los datos residuales y codificar por entropía la información de predicción y los coeficientes de transformación cuantificados para la imagen, tal como se discutió anteriormente con respecto a la Figura 4. Subsecuentemente, el codificador de video 20 puede decodificar la imagen (102), por ejemplo, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir valores residuales, añadiendo luego los valores residuales a los bloques predichos.

La unidad del ALF 64 puede entonces calcular los coeficientes de filtro para varias clases de bloques (104). Por ejemplo, la unidad del ALF 64 puede determinar las clases de bloques con base en gradientes potenciales para los bloques. La unidad del ALF 64 puede entonces generar filtros para las clases de bloques a partir de los coeficientes de filtro calculados (106). La unidad del ALF 64 puede usar cualquiera de las diversas técnicas para generar filtros de esta divulgación. Por ejemplo, para al menos algunas clases de bloques, la unidad del ALF 64 puede seleccionar índices en una pluralidad de conjuntos de coeficientes de filtro calculados, donde el índice para cada clase de bloque identifica el conjunto de coeficientes de filtro calculados que se usarán para generar el filtro para la clase de bloque correspondiente. Adicional o alternativamente, para al menos algunas clases de bloques, la unidad del ALF 64 puede generar datos que indiquen un filtro generado previamente que se usará, así como también una transformación geométrica que se aplicará a al menos una de una región de soporte del filtro o los coeficientes de filtro del filtro generado previamente. Adicional o alternativamente, para al menos algunas clases de bloques, la unidad del ALF 64 puede predecir coeficientes de filtro a partir de un filtro fijo o un filtro de una imagen de referencia, y generar datos que indiquen el filtro fijo o la imagen de referencia a partir de la cual se obtienen los coeficientes de filtro para ser predichos.

Además, el codificador de video 20 puede codificar datos que definen los filtros para las diversas clases de bloques (108). Por ejemplo, para cualquier clase de bloque para el que se van a generar filtros mediante el uso de índices en una pluralidad de conjuntos de coeficientes de filtro, la unidad de codificación por entropía 56 puede codificar por entropía los índices para esas clases de bloques. Adicional o alternativamente, para cualquier clase de bloque para el cual se van a generar filtros mediante el uso de filtros generados previamente y una transformación geométrica, la unidad de codificación por entropía 56 puede codificar por entropía datos que identifiquen los filtros generados previamente y la transformación geométrica que se aplicará (por ejemplo, rotación, volteo en vertical o volteo en diagonal). Adicional o alternativamente, para cualquier clase de bloque para el cual se generen filtros mediante el

uso de coeficientes de filtro predichos de un filtro fijo, la unidad de codificación por entropía 56 puede codificar por entropía datos que indiquen si los coeficientes de filtro deben predecirse a partir de un filtro fijo o un filtro de una imagen de referencia, así como también una indicación del filtro fijo o de la imagen de referencia.

La unidad del ALF 64 también puede filtrar bloques de la imagen decodificada mediante el uso de los filtros correspondientes (110). Es decir, para cada bloque de la imagen, la unidad del ALF 64 puede calcular un gradiente y determinar una clase para el bloque con base en el gradiente. La unidad del ALF 64 puede además seleccionar el filtro correspondiente a la clase. La unidad del ALF 64 puede entonces filtrar cada píxel del bloque mediante el uso del filtro correspondiente a la clase del bloque.

De esta manera, el procedimiento de la Figura 6 representa un ejemplo de un procedimiento de filtrado de un bloque decodificado de datos de video que incluye la construcción de una pluralidad de filtros para clases de bloques de una imagen actual de datos de video, en el que la construcción de la pluralidad de filtros comprende, para cada una de las clases, determinar si un filtro fijo se usa para predecir un conjunto de coeficientes de filtro de la clase, y en respuesta a la determinación de que se usa un filtro fijo para predecir el conjunto de coeficientes de filtro, determinar un valor de índice en un conjunto de filtros fijos de la clase y predecir el conjunto de coeficientes de filtro de la clase mediante el uso de un filtro fijo del conjunto de filtros fijos identificados por el valor de índice, decodificando un bloque actual de una imagen actual de los datos de video, determinando una clase para el bloque actual, seleccionando un filtro de la pluralidad de filtros que corresponde a la clase para el bloque actual, y filtrar al menos un píxel del bloque actual mediante el uso del filtro seleccionado.

La Figura 7 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 6 se describe con respecto al decodificador de video 30 de la Figura 5.

Inicialmente, el decodificador de video 30 decodifica una imagen (120). Por ejemplo, el decodificador de video 30 puede decodificar por entropía la información de predicción y los coeficientes de transformación cuantificados para la imagen, los bloques de intra o inter predicción de la imagen con base en la información de predicción, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir datos residuales, y combinar los bloques predichos y los datos residuales para reproducir los bloques, tal como se discutió anteriormente con respecto a la Figura 5.

La unidad de decodificación por entropía 80 puede además decodificar por entropía datos que definen los filtros para varias clases de bloques (122). Por ejemplo, la unidad de decodificación por entropía 80 puede decodificar por entropía los datos que indican clases de bloques para los cuales se van a generar filtros mediante el uso de índices en una pluralidad de conjuntos de coeficientes de filtro, y decodificar por entropía los índices para esas clases de bloques. Adicional o alternativamente, la unidad de decodificación por entropía 80 puede decodificar por entropía datos que indican clases de bloques para los cuales se generarán filtros mediante el uso de filtros generados previamente y una transformación geométrica, así como también datos que identifican los filtros generados previamente y la transformación geométrica que se aplicará (por ejemplo, rotación, volteo en vertical o volteo en diagonal). Adicional o alternativamente, la unidad de decodificación por entropía 80 puede decodificar por entropía datos que indican clases de bloque para las cuales se generarán filtros mediante el uso de coeficientes de filtro predichos de un filtro fijo, y datos que indican si los coeficientes de filtro deben predecirse a partir de un filtro fijo o de un filtro de una imagen de referencia, así como también una indicación del filtro fijo o de la imagen de referencia.

La unidad del ALF 92 puede entonces calcular los coeficientes de filtro para varias clases de bloques (124). Por ejemplo, la unidad del ALF 92 puede determinar las clases de bloques con base en gradientes potenciales para los bloques. La unidad del ALF 92 puede entonces generar filtros para las clases de bloques a partir de los coeficientes de filtro calculados (126), con base en los datos decodificados de entropía representativos de cómo generar los filtros como se discutió anteriormente. La unidad del ALF 92 puede usar cualquiera de las diversas técnicas para generar filtros de esta divulgación. Por ejemplo, para al menos algunas clases de bloques, la unidad del ALF 92 puede recibir índices de la unidad de decodificación por entropía 80 en una pluralidad de conjuntos de coeficientes de filtro calculados, donde el índice para cada clase de bloque identifica el conjunto de coeficientes de filtro calculados a usar para generar el filtro para la clase de bloque correspondiente. Adicional o alternativamente, para al menos algunas clases de bloques, la unidad del ALF 92 puede recibir datos de la unidad de decodificación por entropía 80 indicando un filtro generado previamente que se usará, así como también una transformación geométrica que se aplicará a al menos una de una región de soporte del filtro o de los coeficientes de filtro del filtro generado previamente. Adicional o alternativamente, para al menos algunas clases de bloques, la unidad del ALF 92 puede predecir coeficientes de filtro a partir de un filtro fijo o un filtro de una imagen de referencia, y generar datos que indiquen el filtro fijo o la imagen de referencia a partir de la cual se obtienen los coeficientes de filtro se predicen, tal como lo indican los datos recibidos de la unidad de decodificación por entropía 80.

La unidad del ALF 92 puede entonces filtrar bloques de la imagen decodificada mediante el uso de los filtros correspondientes (128). Es decir, para cada bloque de la imagen, la unidad del ALF 92 puede calcular un gradiente y determinar una clase para el bloque con base en el gradiente. La unidad del ALF 92 puede además seleccionar el

filtro correspondiente a la clase. La unidad del ALF 92 puede entonces filtrar cada píxel del bloque mediante el uso del filtro correspondiente a la clase del bloque.

De esta manera, el procedimiento de la Figura 7 representa un ejemplo de un procedimiento de filtrado de un bloque decodificado de datos de video que incluye la construcción de una pluralidad de filtros para clases de bloques de una imagen actual de datos de video, en el que la construcción de la pluralidad de filtros comprende, para cada una de las clases, determinar si un filtro fijo se usa para predecir un conjunto de coeficientes de filtro de la clase, y en respuesta a la determinación de que se usa un filtro fijo para predecir el conjunto de coeficientes de filtro, determinar un valor de índice en un conjunto de filtros fijos de la clase y predecir el conjunto de coeficientes de filtro de la clase mediante el uso de un filtro fijo del conjunto de filtros fijos identificados por el valor de índice, decodificando un bloque actual de una imagen actual de los datos de video, determinando una clase para el bloque actual, seleccionando un filtro de la pluralidad de filtros que corresponde a la clase para el bloque actual, y filtrar al menos un píxel del bloque actual mediante el uso del filtro seleccionado.

La Figura 8 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 8 se describe con respecto al codificador de video 20 de la Figura 4.

Inicialmente, el codificador de video 20 codifica una imagen (140). Por ejemplo, el codificador de video 20 puede intra o inter predecir los bloques de la imagen, calcular los residuales para los bloques predichos, transformar y cuantificar los datos residuales y codificar por entropía la información de predicción y los coeficientes de transformación cuantificados para la imagen, tal como se discutió anteriormente con respecto a la Figura 4. Subsecuentemente, el codificador de video 20 puede decodificar la imagen (142), por ejemplo, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir valores residuales, añadiendo luego los valores residuales a los bloques predichos.

La unidad del ALF 64 puede entonces seleccionar un filtro (tal como de acuerdo con el filtrado de bucle adaptativo (ALF)) para una clase de un bloque de la imagen (144). La unidad del ALF 64 puede seleccionar la clase en base a, por ejemplo, un gradiente calculado para uno o más píxeles del bloque. Alternativamente, la unidad del ALF 64 puede determinar una clase para el bloque con base en otras características del bloque.

De acuerdo con las técnicas de esta divulgación, la unidad del ALF 64 puede calcular un gradiente del bloque (146). La unidad del ALF 64 puede además determinar y realizar una transformación geométrica con base en el gradiente calculado (148). Por ejemplo, la unidad del ALF 64 puede determinar una transformación geométrica de acuerdo con las técnicas de la Tabla 1 como se discutió anteriormente. La unidad del ALF 64 puede aplicar la transformación geométrica a los coeficientes del filtro seleccionado o a una región de soporte del filtro del filtro seleccionado (por ejemplo, píxeles vecinos a un píxel a filtrar mediante el uso del filtro seleccionado). La unidad del ALF 64 puede entonces filtrar el bloque de la imagen decodificada mediante el uso del filtro seleccionado (150). Es decir, la unidad del ALF 64 filtra cada píxel del bloque mediante el uso del filtro seleccionado.

De esta manera, el procedimiento de la Figura 8 representa un ejemplo de un procedimiento para filtrar un bloque decodificado de datos de video que incluye decodificar un bloque actual de una imagen actual de los datos de video, seleccionar un filtro (tal como de acuerdo con el filtrado de bucle adaptativo (ALF)) que se usará para filtrar los píxeles del bloque actual, seleccionar una transformación geométrica que se realizará en una de una región de soporte del filtro o de los coeficientes del filtro seleccionado, realizar la transformación geométrica en la región de soporte del filtro o en los coeficientes del filtro seleccionado, y filtrar el al menos un píxel del bloque actual mediante el uso del filtro seleccionado después de realizar la transformación geométrica.

La Figura 9 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 9 se describe con respecto al decodificador de video 30 de la Figura 5.

Inicialmente, el decodificador de video 30 decodifica una imagen (160), por ejemplo, por valores de decodificación por entropía de elementos de sintaxis para la predicción y los coeficientes de transformación cuantificados, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir valores residuales, añadiendo luego los valores residuales a los bloques de predicción. La unidad del ALF 92 puede entonces seleccionar un filtro (tal como de acuerdo con el filtrado de bucle adaptativo (ALF)) para una clase de un bloque de la imagen (162). La unidad del ALF 92 puede seleccionar la clase en base a, por ejemplo, un gradiente calculado para uno o más píxeles del bloque. Alternativamente, la unidad del ALF 92 puede determinar una clase para el bloque con base en otras características del bloque.

De acuerdo con las técnicas de esta divulgación, la unidad del ALF 92 puede calcular un gradiente del bloque (164). La unidad del ALF 92 puede además determinar y realizar una transformación geométrica con base en el gradiente calculado (166). Por ejemplo, la unidad del ALF 92 puede determinar una transformación geométrica de acuerdo con

las técnicas de la Tabla 1 como se discutió anteriormente. La unidad del ALF 92 puede aplicar la transformación geométrica a los coeficientes del filtro seleccionado o a una región de soporte del filtro del filtro seleccionado (por ejemplo, píxeles vecinos a un píxel a filtrar mediante el uso del filtro seleccionado). La unidad del ALF 92 puede entonces filtrar el bloque de la imagen decodificada mediante el uso del filtro seleccionado (168). Es decir, la unidad del ALF 92 filtra cada píxel del bloque mediante el uso del filtro seleccionado.

De esta manera, el procedimiento de la Figura 9 representa un ejemplo de un procedimiento para filtrar un bloque decodificado de datos de video que incluye decodificar un bloque actual de una imagen actual de los datos de video, seleccionar un filtro (tal como de acuerdo con el filtrado de bucle adaptativo (ALF)) que se usará para filtrar los píxeles del bloque actual, seleccionar una transformación geométrica que se realizará en una de una región de soporte del filtro o de los coeficientes del filtro seleccionado, realizar la transformación geométrica en la región de soporte del filtro o en los coeficientes del filtro seleccionado, y filtrar el al menos un píxel del bloque actual mediante el uso del filtro seleccionado después de realizar la transformación geométrica.

La Figura 10 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un codificador de video (por ejemplo, durante un proceso de codificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 10 se describe con respecto al codificador de video 20 de la Figura 4.

Inicialmente, el codificador de video 20 codifica una imagen (170). Por ejemplo, el codificador de video 20 puede intra o inter predecir los bloques de la imagen, calcular los residuales para los bloques predichos, transformar y cuantificar los datos residuales y codificar por entropía la información de predicción y los coeficientes de transformación cuantificados para la imagen, tal como se discutió anteriormente con respecto a la Figura 4. Subsecuentemente, el codificador de video 20 puede decodificar la imagen (172), por ejemplo, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir valores residuales, añadiendo luego los valores residuales a los bloques predichos.

La unidad del ALF 64 puede entonces predecir los coeficientes de un filtro (tal como de acuerdo con el filtrado de bucle adaptativo (ALF)) para una clase de bloque de la imagen mediante el uso de un filtro fijo (174). La unidad del ALF 64 puede seleccionar la clase en base a, por ejemplo, un gradiente calculado para uno o más píxeles del bloque. Alternativamente, la unidad del ALF 64 puede determinar una clase para el bloque con base en otras características del bloque. En algunos ejemplos, pueden estar disponibles varios filtros fijos para cada clase de bloque, y la unidad del ALF 64 puede probar cada uno de los filtros fijos disponibles para determinar cuál da como resultado el mejor rendimiento de distorsión de velocidad. La unidad del ALF 64 puede entonces seleccionar los filtros fijos que produzcan el mejor rendimiento de distorsión de velocidad para cada clase de bloque, o un filtro de referencia de una imagen de referencia si el filtro de referencia produce un mejor rendimiento de distorsión de velocidad. La unidad del ALF 64 codifica entonces una bandera que indica cuál de los filtros fijos se usa para cada clase de bloque (y adicional o alternativamente, si el filtro para una clase particular se predice a partir de un filtro de referencia de una imagen de referencia) (176).

La unidad del ALF 64 puede entonces seleccionar filtros para cada clase de bloque de la imagen decodificada (178), determinar clases para cada uno de los bloques de la imagen decodificada (180), por ejemplo, con base en los gradientes calculados para los bloques, y filtrar los bloques mediante el uso de los filtros seleccionados (182). Una vez más, la unidad del ALF 64 puede probar diferentes filtros para cada clase de bloque para determinar qué filtro produce el mejor rendimiento de distorsión de velocidad. De acuerdo con las técnicas de esta divulgación, en algunos ejemplos, la unidad del ALF 64 puede fusionar dos o más clases de bloques, de manera que cada una de las clases fusionadas use el mismo filtro. Además, el codificador de video 20 puede codificar datos que indiquen cuál de los filtros se usará para cada clase de bloque.

De esta manera, el procedimiento de la Figura 10 representa un ejemplo de un procedimiento de filtrado de un bloque decodificado de datos de video que incluye la construcción de una pluralidad de filtros para clases de bloques de una imagen actual de datos de video, en el que la construcción de una pluralidad de filtros para clases de bloques de una imagen actual de datos de video, en el que la construcción de la pluralidad de filtros comprende, para cada una de las clases, determinar si se usa un filtro fijo para predecir un conjunto de coeficientes de filtro de la clase, y en respuesta a determinar que se usa un filtro fijo para predecir el conjunto de filtros coeficientes, determinando un valor de índice en un conjunto de filtros fijos de la clase y prediciendo el conjunto de coeficientes de filtro de la clase mediante el uso de un filtro fijo del conjunto de filtros fijos identificados por el valor de índice, decodificando un bloque actual de una imagen actual de los datos de video, determinando una clase para el bloque actual, seleccionando un filtro de la pluralidad de filtros que corresponde a la clase para el bloque actual, y filtrando al menos un píxel del bloque actual mediante el uso del filtro seleccionado.

La Figura 11 es un diagrama de flujo que ilustra un procedimiento ilustrativo para filtrar bloques de una imagen decodificada por un decodificador de video (por ejemplo, durante un proceso de decodificación de video). Con fines ilustrativo y de explicación, el procedimiento de la Figura 11 se describe con respecto al decodificador de video 30 de la Figura 5.

Inicialmente, el decodificador de video 30 decodifica una imagen de datos de video (190), por ejemplo, por valores de decodificación por entropía de elementos de sintaxis para la predicción y los coeficientes de transformación cuantificados, cuantificando y transformando inversamente los coeficientes de transformación cuantificados para reproducir valores residuales, añadiendo luego los valores residuales a los bloques predichos.

La unidad del ALF 92 puede decodificar entonces una bandera que indica si los coeficientes de un filtro deben predecirse a partir de un filtro fijo (192) y, en algunos ejemplos, si los coeficientes de filtro deben predecirse a partir de un filtro fijo, datos que identifican el filtro fijo. La unidad del ALF 92 puede entonces predecir los coeficientes del filtro mediante el uso del filtro fijo (asumiendo que los coeficientes del filtro deben predecirse a partir del filtro fijo en este ejemplo) (194). En otros ejemplos, la unidad del ALF 92 puede predecir coeficientes de filtro a partir de un filtro de referencia de una imagen de referencia, si la bandera indica que los coeficientes de filtro no se predicen a partir de un filtro fijo.

La unidad del ALF 92 puede entonces seleccionar filtros para cada clase de bloque de la imagen decodificada (196), determinar clases para cada uno de los bloques de la imagen decodificada (198), por ejemplo, con base en los gradientes calculados para los bloques, y filtrar los bloques mediante el uso de los filtros seleccionados (200). La unidad del ALF 92 puede decodificar datos para cada clase de bloque identificando un filtro que se usará para filtrar los píxeles de bloques de esa clase.

De esta manera, el procedimiento de la Figura 11 representa un ejemplo de un procedimiento de filtrado de un bloque decodificado de datos de video que incluye la construcción de una pluralidad de filtros para clases de bloques de una imagen actual de datos de video, en el que la construcción de la pluralidad de filtros comprende, para cada una de las clases, determinar si un filtro fijo se usa para predecir un conjunto de coeficientes de filtro de la clase, y en respuesta a la determinación de que se usa un filtro fijo para predecir el conjunto de coeficientes de filtro, determinar un valor de índice en un conjunto de filtros fijos de la clase y predecir el conjunto de coeficientes de filtro de la clase mediante el uso de un filtro fijo del conjunto de filtros fijos identificados por el valor de índice, decodificando un bloque actual de una imagen actual de los datos de video, determinando una clase para el bloque actual, seleccionando un filtro de la pluralidad de filtros que corresponde a la clase para el bloque actual, y filtrar al menos un píxel del bloque actual mediante el uso del filtro seleccionado.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprogramas o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse o transmitirse, tal como una o más instrucciones o código, en un medio legible por ordenador y ejecutadas por una unidad de procesamiento con base en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como medios de almacenamiento de datos, o medios de comunicación, incluido cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles 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 pueda accederse mediante uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A manera de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y a las que puede accederse mediante un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante el uso de un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas tal como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas tal 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 están dirigidos a medios de almacenamiento tangibles no transitorios. El disco, tal como se usa en la presente memoria, incluye disco compacto (CD), disco de láser, disco óptico, disco digital versátil (DVD), disquete, y disco de Blu-ray donde existen discos que usualmente reproducen magnéticamente los datos, mientras que otros discos reproducen ópticamente los datos con láseres. Las combinaciones de los medios anteriores pueden incluirse además dentro del ámbito de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, tal como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices lógicas programables en campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", tal como se usa en la presente memoria, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la



presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un código combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos. Por lo tanto, un procesador puede formarse por cualquiera de una variedad de circuitos de procesamiento integrados que comprenden uno o más procesadores implementados como circuitos de procesamiento de hardware fijo, circuitos de procesamiento programables y/o una combinación de circuitos de procesamiento tanto fijos como programables.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, tal como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de código o proporcionarse mediante una colección de unidades de hardware interoperativo, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del ámbito de las siguientes reivindicaciones.

## REIVINDICACIONES

1. Un procedimiento para filtrar un bloque decodificado de datos de video, comprendiendo el método:

5 decodificar (160, 140) un bloque actual de una imagen actual de los datos de video;  
seleccionar (162, 144) un filtro que se usará para filtrar al menos un píxel del bloque actual en el que  $f(k,l)$   
representa el filtro seleccionado y  $R(i, j)$  representa un píxel en la posición  $(i, j)$  de la imagen actual;  
seleccionar una transformación geométrica que se realizará en una región de soporte del filtro o de los  
10 coeficientes del filtro seleccionado, en el que la transformación geométrica comprende una transformación de  
rotación, una transformación de volteo en diagonal o de volteo en vertical, y en el que la selección de la  
transformación geométrica comprende:

calcular un gradiente horizontal  $g_h$  de acuerdo con:

$$15 \quad g_h = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} H_{k,l},$$

donde  $H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|$ ;

calcular un gradiente vertical  $g_v$  de acuerdo con:

$$20 \quad g_v = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} V_{k,l},$$

donde  $V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|$ ;

25 calcular un primer gradiente diagonal  $g_{d1}$  de acuerdo con:

$$g_{d1} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D1_{k,l},$$

30 donde  $D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$ ;

calcular un segundo gradiente diagonal  $g_{d2}$  de acuerdo con:

$$35 \quad g_{d2} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D2_{k,l},$$

donde  $D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$ ;

seleccionar la transformación de volteo en diagonal cuando  $g_{d2} < g_{d1}$  y  $g_v < g_h$ ;

seleccionar la transformación de volteo en vertical cuando  $g_{d1} < g_{d2}$  y  $g_h < g_v$ ; o

seleccionar la transformación de rotación cuando  $g_{d1} < g_{d2}$  y  $g_v < g_h$ ;

40 realizar (166, 148) la transformación geométrica en la región de soporte del filtro o en los coeficientes del filtro  
seleccionado; y

filtrar (168, 150) el al menos un píxel del bloque actual mediante el uso del filtro seleccionado después de  
realizar la transformación geométrica.

45 2. El procedimiento de la reivindicación 1,

en el que la transformación de rotación comprende  $f_R(k,l) = f(K-l-1, k)$ ,

en el que la transformación de volteo en diagonal comprende  $f_D(k, l) = f(l, k)$ ,

50 en el que la transformación de volteo en vertical comprende  $f_V(k, l) = f(k, K-l-1)$ , y

en el que K es el tamaño del filtro seleccionado, k y l son coordenadas de coeficientes del filtro seleccionado  
o coordenadas de valores en la región de soporte del filtro,  $0 \leq k, l \leq K-1$ , la ubicación (0, 0) está en la esquina  
superior izquierda del filtro seleccionado o la región de soporte del filtro, y la ubicación (K-1, K-1) está en la  
esquina inferior derecha de la esquina superior izquierda del filtro seleccionado o la región de soporte del  
filtro.

55 3. El procedimiento de la reivindicación 1, en el que la región de soporte del filtro comprende una pluralidad de  
píxeles vecinos a al menos un píxel del bloque actual al que se aplicarán los coeficientes del filtro seleccionado, y  
filtrar el al menos un píxel comprende realizar la transformación geométrica en la región de soporte del filtro o en  
los coeficientes del filtro seleccionado.

60 4. El procedimiento de la reivindicación 1, que comprende además codificar el bloque actual antes de decodificar el  
bloque actual.

5. Un dispositivo para filtrar un bloque descodificado de datos de video, comprendiendo el dispositivo:

65 una memoria configurada para almacenar los datos de video; y

uno o más procesadores implementados en circuitos y configurados para:

decodificar un bloque actual de una imagen actual de los datos de video;

seleccionar un filtro que se usará para filtrar al menos un píxel del bloque actual en el que  $f(k,l)$  representa el filtro seleccionado y  $R(i,j)$  representa un píxel en la posición  $(i,j)$  de la imagen actual;

seleccionar una transformación geométrica que se realizará en una región de soporte del filtro o en los coeficientes del filtro seleccionado, en el que la transformación geométrica comprende una transformación de rotación, una transformación de volteo en diagonal o de volteo en vertical, y en el que el uno o más procesadores se configuraran para seleccionar la geometría mediante:

calcular un gradiente horizontal  $g_h$  de acuerdo con:

$$g_h = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} H_{k,l},$$

donde  $H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|$ ;

calcular un gradiente vertical  $g_v$  de acuerdo con:

$$g_v = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} V_{k,l},$$

donde  $V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|$ ;

calcular un primer gradiente diagonal  $g_{d1}$  de acuerdo con:

$$g_{d1} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D1_{k,l},$$

donde  $D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$ ;

calcular un segundo gradiente diagonal  $g_{d2}$  de acuerdo con:

$$g_{d2} = \sum_{k=i-M}^{i+N+M-1} \sum_{l=j-M}^{j+N+M-1} D2_{k,l},$$

donde  $D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$ ;

seleccionar la transformación de volteo en diagonal cuando  $g_{d2} < g_{d1}$  y  $g_v < g_h$ ;

seleccionar la transformación de volteo en vertical cuando  $g_{d1} < g_{d2}$  y  $g_h < g_v$ ; o

seleccionar la transformación de rotación cuando  $g_{d1} < g_{d2}$  y  $g_v < g_h$ ;

realizar la transformación geométrica en la región de soporte del filtro o en los coeficientes del filtro seleccionado; y

filtrar al menos un píxel del bloque actual mediante el uso del filtro seleccionado después de realizar la transformación geométrica.

6. El dispositivo de la reivindicación 5, en el que el dispositivo es un dispositivo de comunicación inalámbrica, que comprende además:

un receptor configurado para recibir datos de video, que incluye la imagen actual.

7. El dispositivo de la reivindicación 6, en el que el dispositivo de comunicación inalámbrica es un teléfono celular y los datos de video se reciben por el receptor y se modulan de acuerdo con un estándar de comunicación celular.

8. Un medio de almacenamiento legible por ordenador que tiene almacenadas en él instrucciones que, cuando se ejecutan, hacen que un procesador ejecute el procedimiento de cualquiera de las reivindicaciones 1 a 4.

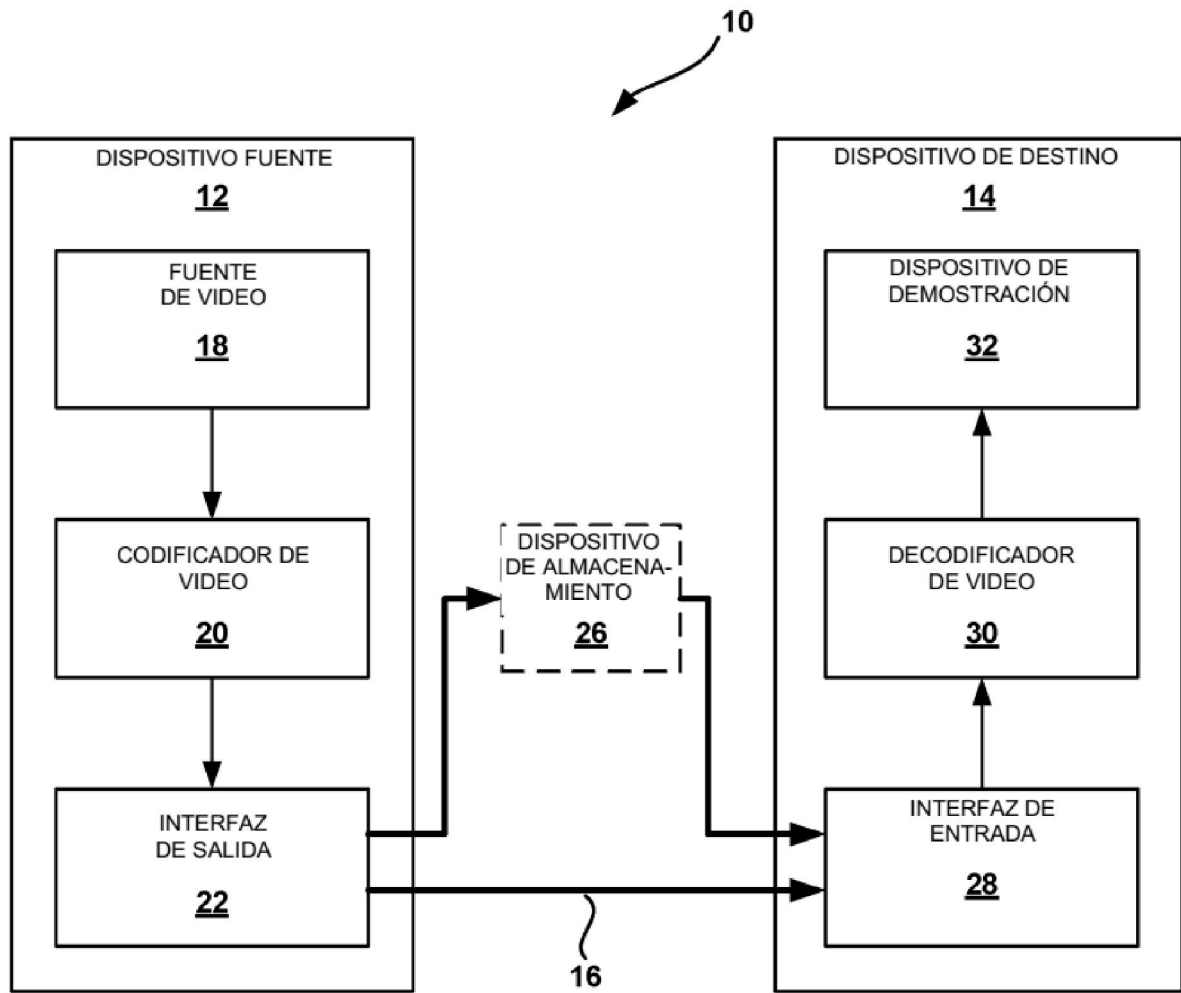


Figura 1

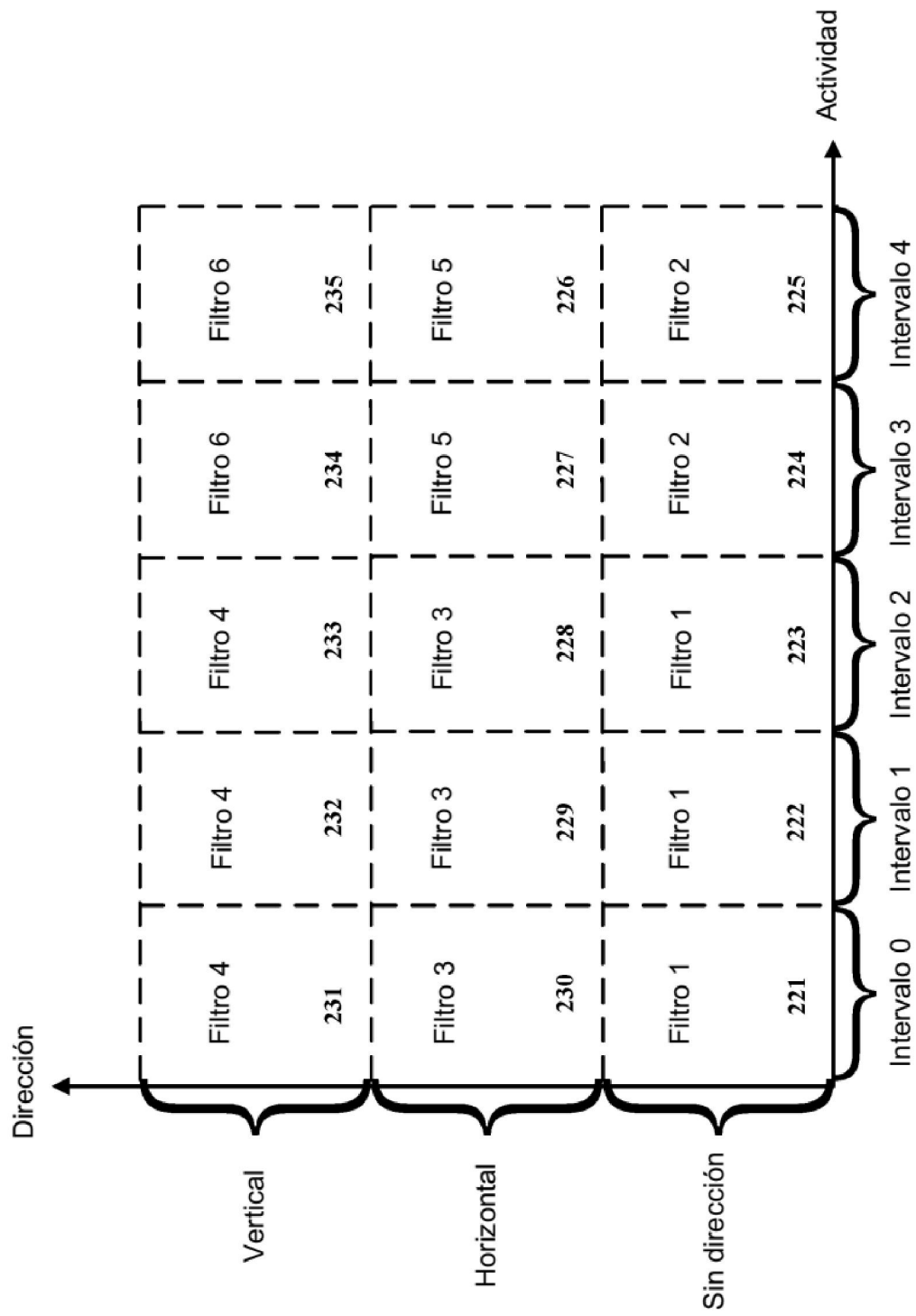


Figura 2

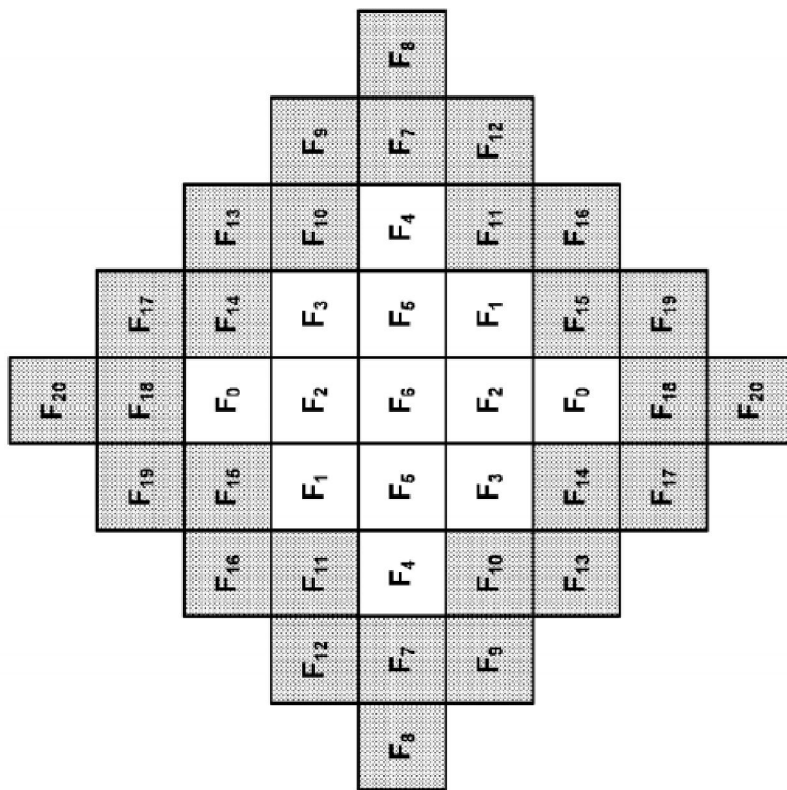


Figura 3

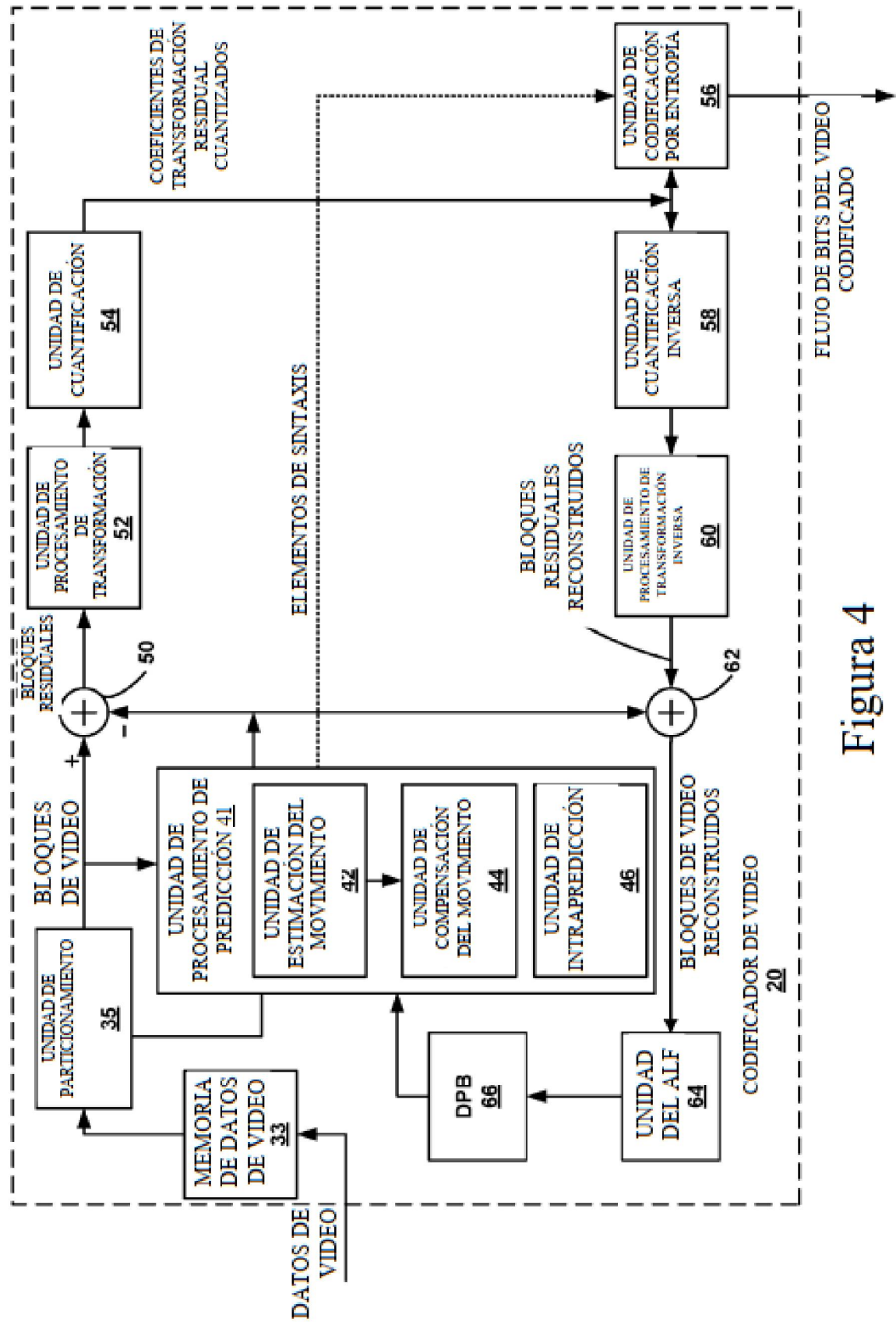


Figura 4

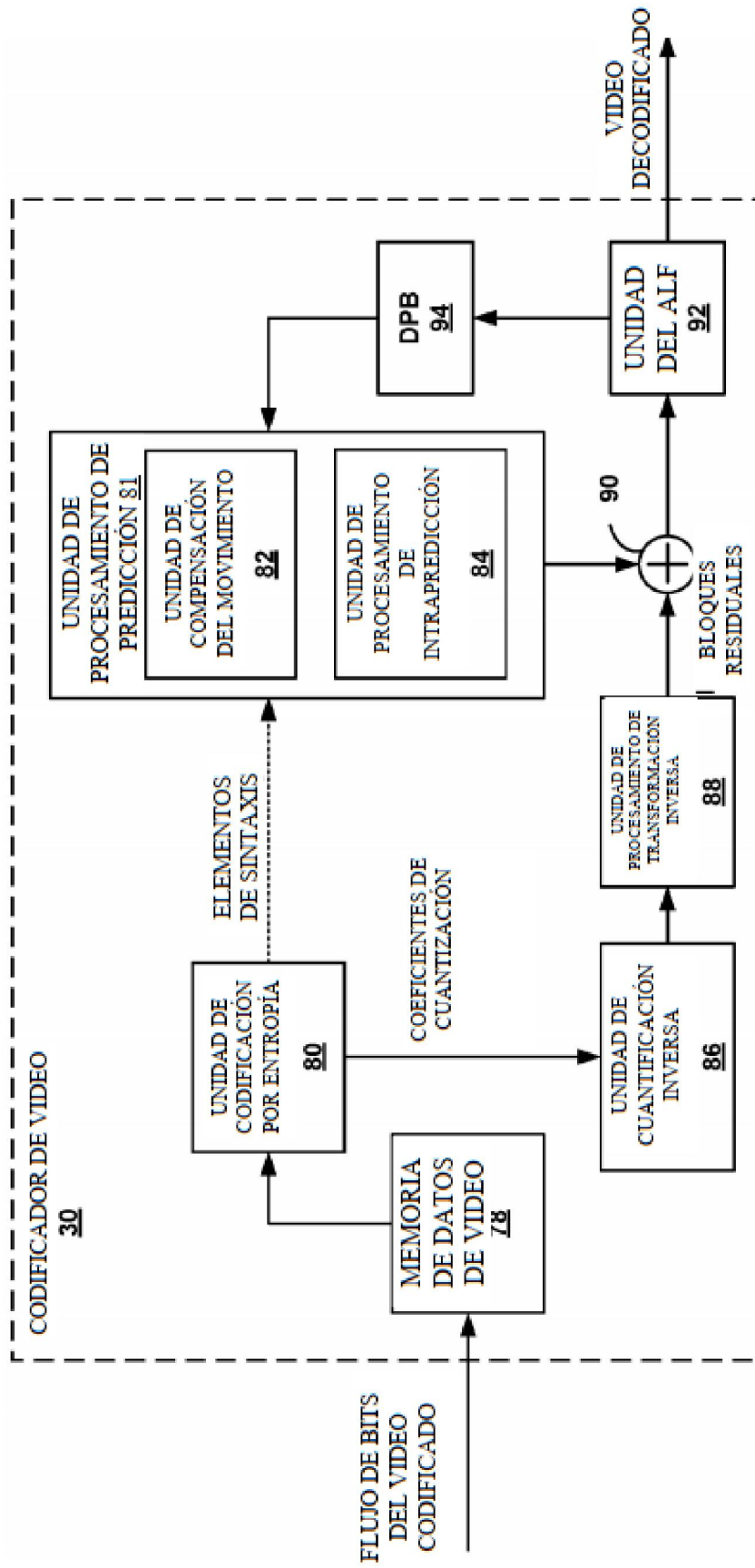


Figura 5



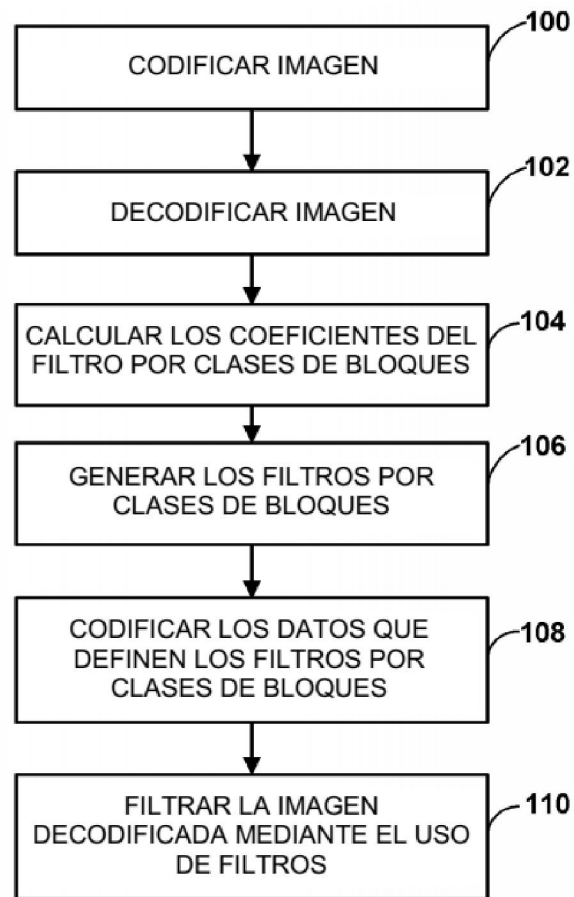


Figura 6

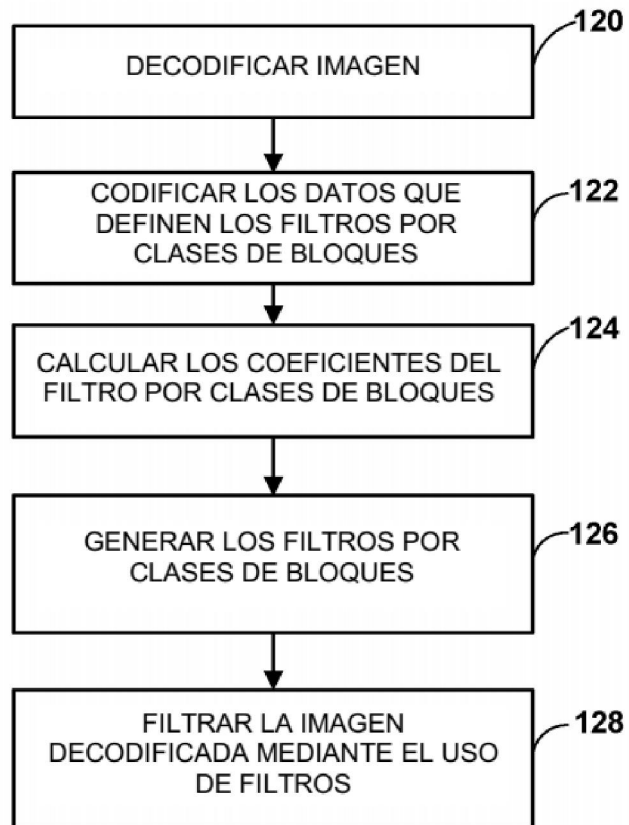


Figura 7

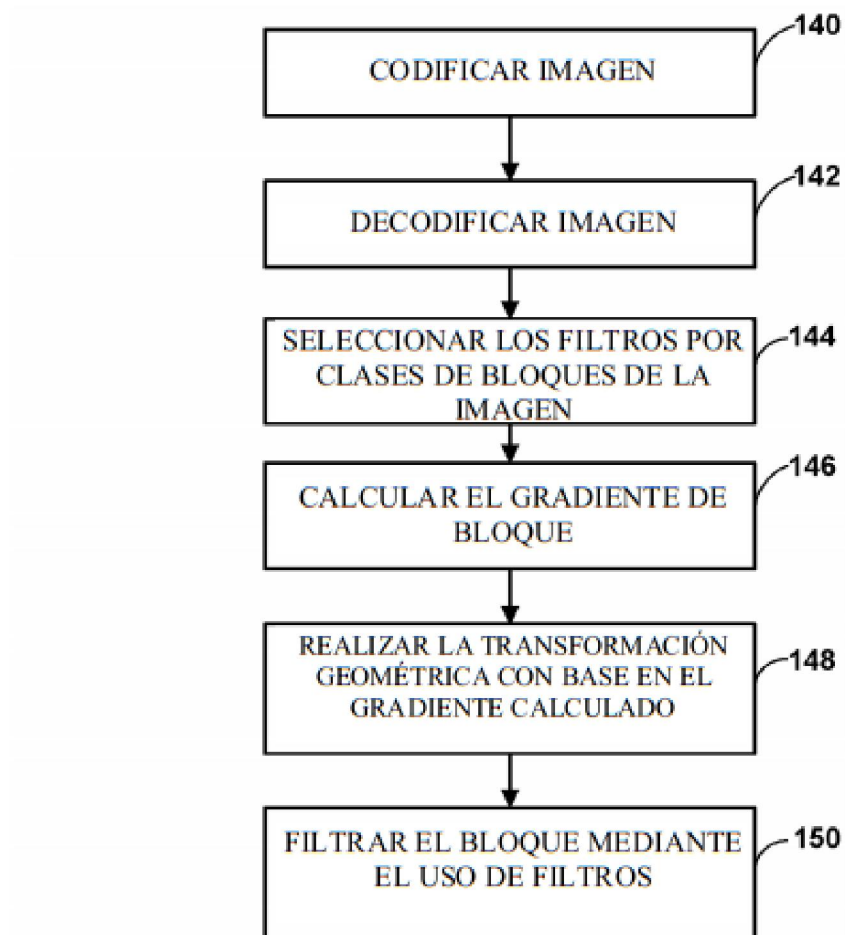


Figura 8

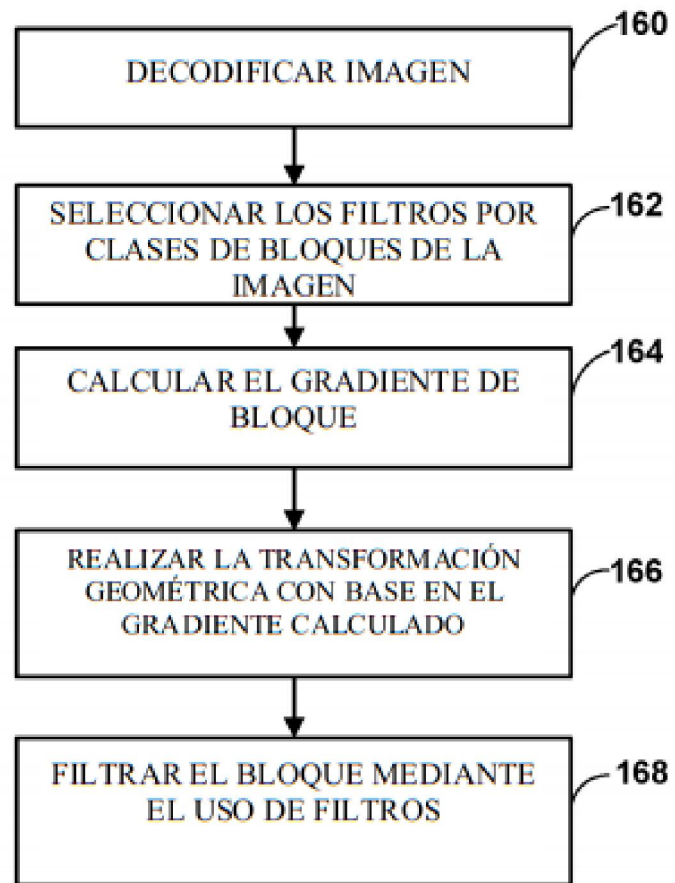


Figura 9

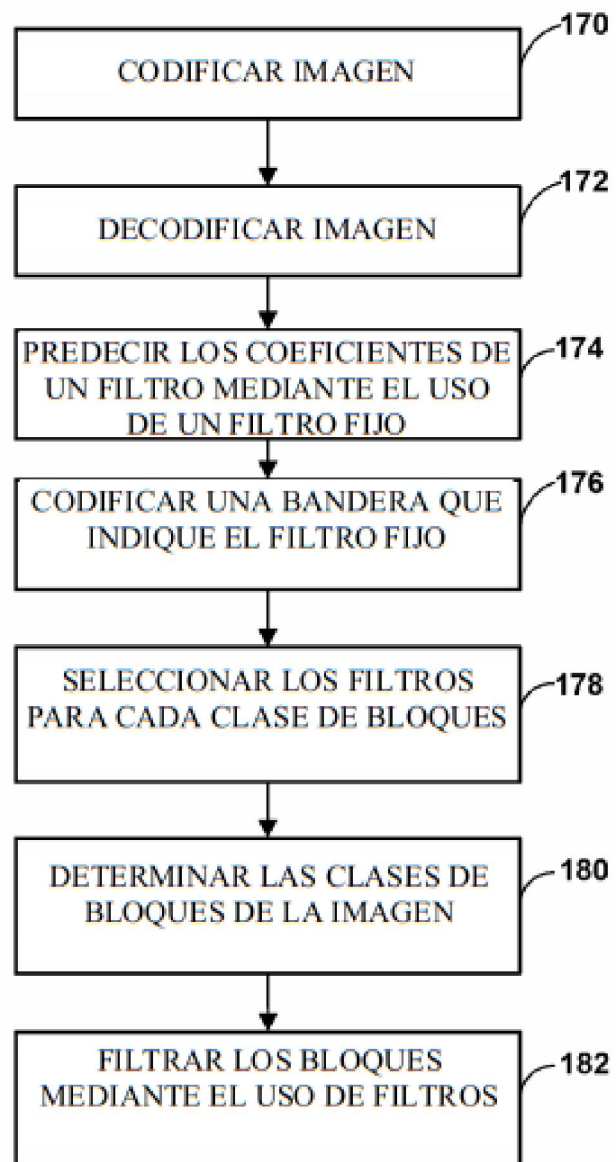


Figura 10

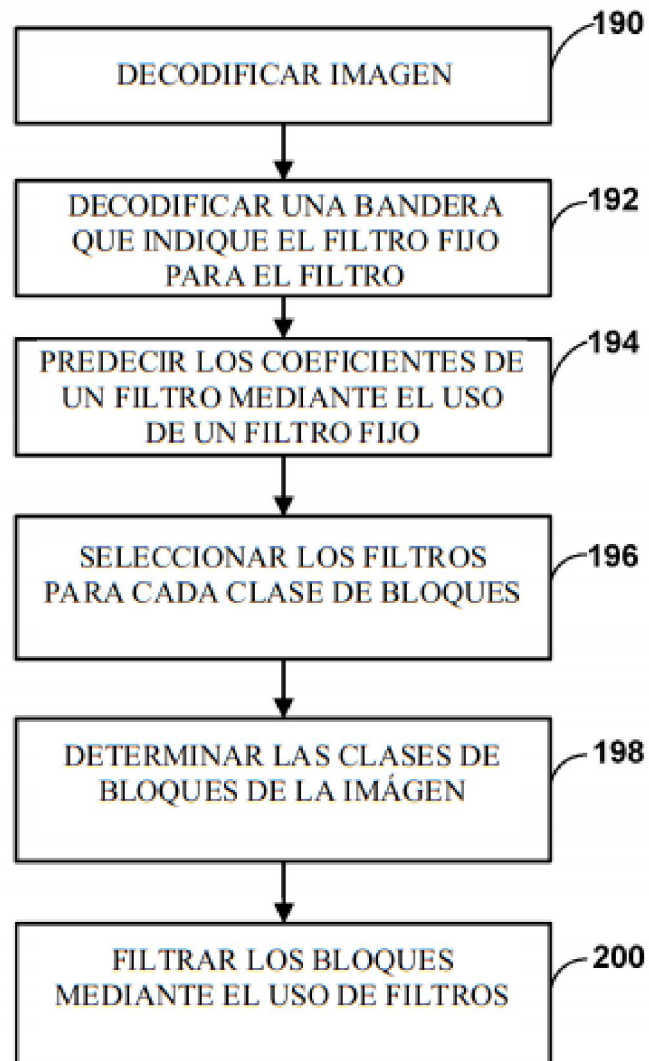


Figura 11