

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 824 831**

51 Int. Cl.:

H04N 19/117 (2014.01)
H04N 19/176 (2014.01)
H04N 19/70 (2014.01)
H04N 19/46 (2014.01)
H04N 19/14 (2014.01)
H04N 19/182 (2014.01)
H04N 19/82 (2014.01)
H04N 19/42 (2014.01)
H04N 19/85 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **22.02.2012** **PCT/US2012/026165**
87 Fecha y número de publicación internacional: **30.08.2012** **WO12116094**
96 Fecha de presentación y número de la solicitud europea: **22.02.2012** **E 12706180 (2)**
97 Fecha y número de publicación de la concesión europea: **08.07.2020** **EP 2679009**

54 Título: **Filtrado multimétrico**

30 Prioridad:

23.02.2011 US 201161445967 P
03.03.2011 US 201161448771 P
08.04.2011 US 201161473713 P
16.04.2011 US 201161476260 P
22.04.2011 US 201161478287 P
30.06.2011 US 201161503426 P
30.06.2011 US 201161503434 P
30.06.2011 US 201161503440 P
25.08.2011 US 201161527463 P
06.09.2011 US 201161531571 P
21.02.2012 US 201213401573

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
13.05.2021

73 Titular/es:

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

72 Inventor/es:

CHONG, IN SUK y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 824 831 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

Filtrado multimétrico

CAMPO TÉCNICO

[0001] Esta divulgación se refiere a la codificación de vídeo digital basada en bloques usada para comprimir datos de vídeo y, más en particular, a técnicas para el filtrado de bloques de vídeo.

[0002] El documento de CHEN ET AL: "JCT-VC MEETING; 20-1-2011 [CONFERENCIA DE JCT-VC; 20/1/2011]" describe varias mejoras en filtros de bucle adaptativos. El documento de HUANG ET AL: "JCT-VC MEETING; 21.7.2010 [CONFERENCIA DE JCT-VC; 21/7/2010]" describe técnicas relacionadas con la restauración adaptativa en bucle.

ANTECEDENTES

[0003] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, dispositivos de comunicación inalámbrica, tales como radiotelefonos, sistemas de radiodifusión inalámbrica, asistentes personales digitales (PDA), ordenadores portátiles, ordenadores de escritorio, tabletas electrónicas, cámaras digitales, dispositivos de grabación digital, dispositivos de videojuegos, consolas de videojuegos y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como MPEG-2, MPEG-4 o H.264/MPEG-4 de UIT-T, parte 10, codificación avanzada de vídeo (AVC), para transmitir y recibir vídeo digital más eficazmente. Las técnicas de compresión de vídeo realizan predicción espacial y temporal para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Nuevas normas de vídeo, tales como la norma de codificación de vídeo de alta eficacia (HEVC) que se está desarrollando por el "Equipo de Colaboración Conjunta de Codificación de Vídeo" (JCTVC), que es una colaboración entre MPEG e UIT-T, continúan surgiendo y evolucionando. Esta nueva norma HEVC también se denomina a veces H.265.

[0004] Las técnicas de compresión de vídeo basadas en bloques pueden realizar predicción espacial y/o predicción temporal. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial entre bloques de vídeo dentro de una unidad de vídeo codificado dada, que puede comprender una trama de vídeo, un fragmento de una trama de vídeo o similares. Por el contrario, la intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal entre bloques de vídeo de unidades de codificación sucesivas de una secuencia de vídeo. Para la intracodificación, un codificador de vídeo realiza una predicción espacial para comprimir datos en base a otros datos dentro de la misma unidad de vídeo codificado. Para la intercodificación, el codificador de vídeo realiza una estimación de movimiento y compensación de movimiento para rastrear el movimiento de los bloques de vídeo correspondientes de dos o más unidades de vídeo codificado adyacentes.

[0005] Un bloque de vídeo codificado se puede representar por información de predicción que se puede usar para crear o identificar un bloque predictivo, y un bloque residual de datos indicativos de diferencias entre el bloque que se está codificando y el bloque predictivo. En el caso de la intercodificación, se usan uno o más vectores de movimiento para identificar el bloque predictivo de datos de una unidad de codificación previa o posterior, mientras que en el caso de la intracodificación, el modo de predicción se puede usar para generar el bloque predictivo en base a datos dentro de la CU asociada al bloque de vídeo que se está codificando. Tanto la intracodificación como la intercodificación pueden definir varios modos de predicción diferentes, que pueden definir diferentes tamaños de bloque y/o técnicas de predicción usadas en la codificación. También se pueden incluir tipos adicionales de elementos de sintaxis como parte de los datos de vídeo codificados para controlar o definir las técnicas o parámetros de codificación usados en el proceso de codificación.

[0006] Después de la codificación de predicción basada en bloques, el codificador de vídeo puede aplicar procesos de transformada, cuantificación y codificación por entropía para reducir adicionalmente la velocidad de transmisión de bits asociada a la comunicación de un bloque residual. Las técnicas de transformada pueden comprender procesos de transformadas de coseno discretas (DCT) o procesos conceptualmente similares, tales como transformadas de ondículas, transformadas enteras u otros tipos de transformadas. En un proceso de transformada de coseno discreta, como ejemplo, el proceso de transformada convierte un conjunto de valores de diferencia de píxeles en coeficientes de transformada, que pueden representar la energía de los valores de píxel en el dominio de frecuencia. La cuantificación se aplica a los coeficientes de transformada, y en general implica un proceso que limita el número de bits asociados a cualquier coeficiente de transformada dado. La codificación por entropía comprende uno o más procesos que comprimen conjuntamente una secuencia de coeficientes de transformada cuantificados.

[0007] El filtrado de bloques de vídeo se puede aplicar como parte de los bucles de codificación y decodificación, o como parte de un proceso posterior al filtrado en bloques de vídeo reconstruidos. El filtrado se usa comúnmente, por ejemplo, para reducir el efecto pixelado u otros artefactos comunes a la codificación de vídeo basada en bloques. Los coeficientes de filtro (a veces denominados derivaciones de filtro) se pueden definir o seleccionar

para promover niveles deseables de filtrado de bloques de vídeo que puedan reducir el efecto pixelado y/o mejorar la calidad de vídeo de otras maneras. Un conjunto de coeficientes de filtro, por ejemplo, puede definir cómo se aplica el filtrado a lo largo de los bordes de los bloques de vídeo o de otras localizaciones dentro de los bloques de vídeo. Diferentes coeficientes de filtro pueden provocar diferentes niveles de filtrado con respecto a diferentes píxeles de los bloques de vídeo. El filtrado, por ejemplo, puede suavizar o agudizar las diferencias en intensidad de los valores de píxel adyacentes para ayudar a eliminar los artefactos no deseados. Se llama la atención al documento Y-W HUANG ET AL: "In-loop adaptive restoration [Restauración adaptativa en bucle]", 2. CONVENCION DE JCT-VC; 21-7-2010 - 28-7-2010; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA SOBRE LA CODIFICACIÓN DE VÍDEO DE ISO/CEI JTC1/SC29/WG11 Y UIT-T SG.16); URL:HTTP://WFTP3.1TU.INT/AV-ARCH/JCTVC-SITE/ n.º JCTVC-8077, 25 de julio de 2010 (25-07-2010), ISSN: 0000-0046.

BREVE EXPLICACIÓN

[0008] La invención se expone en el juego de reivindicaciones adjuntas. La presente divulgación describe técnicas asociadas al filtrado de datos de vídeo en un proceso de codificación de vídeo y/o decodificación de vídeo. De acuerdo con la presente divulgación, el filtrado se aplica en un codificador, y la información del filtro se codifica en el flujo de bits para posibilitar que un decodificador identifique el filtrado que se aplicó en el codificador. El decodificador recibe datos de vídeo codificados que incluyen la información del filtro, decodifica los datos de vídeo y aplica el filtrado en base a la información de filtrado. De esta manera, el decodificador aplica el mismo filtrado que se aplicó en el codificador. De acuerdo con las técnicas de la presente divulgación, de trama en trama, de fragmento en fragmento o de LCU en LCU, un codificador puede seleccionar uno o más conjuntos de filtros, y de unidad codificada en unidad codificada, el codificador puede determinar si se aplica o no el filtrado. Para las unidades codificadas (CU) que se van a filtrar, el codificador puede realizar el filtrado de píxel en píxel o de grupo en grupo, donde un grupo, por ejemplo, podría ser un bloque de píxeles 2x2 o un bloque de píxeles 4x4.

[0009] En un ejemplo, un procedimiento de decodificación de datos de vídeo incluye generar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; en respuesta a la recepción de una primera contraseña, donde la primera contraseña señala que una combinación de rango actual se asigna a un mismo filtro que una combinación de rango previa, asignar la combinación de rango actual al mismo filtro; y, en respuesta a la recepción de una segunda contraseña, donde la segunda contraseña señala que la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, asignar la combinación de rango actual a un nuevo filtro.

[0010] En otro ejemplo, un aparato para codificar datos de vídeo incluye un decodificador de vídeo configurado para generar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; asignar una combinación de rango actual a un mismo filtro que una combinación de rango previa en respuesta a la recepción de una primera contraseña que señala que la combinación de rango actual está asignada al mismo filtro que la combinación de rango previa; y asignar la combinación de rango actual a un filtro identificado por una segunda contraseña en respuesta a la recepción de la segunda contraseña que señala que la combinación de rango actual está asignada a un filtro diferente que la combinación de rango previa.

[0011] En otro ejemplo, un aparato incluye medios para generar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; medios para asignar la combinación de rango actual a un mismo filtro que una combinación de rango previa en respuesta a la recepción de una primera contraseña que señala que la combinación de rango actual está asignada al mismo filtro que la combinación de rango previa; y medios para asignar la combinación de rango actual a un filtro identificado por una segunda contraseña en respuesta a la recepción de la segunda contraseña que señala que la combinación de rango actual está asignada a un filtro diferente que la combinación de rango previa.

[0012] En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que cuando se ejecutan hacen que o más procesadores generen una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; asignen una combinación de rango actual a un mismo filtro que una combinación de rango previa en respuesta a la recepción de una primera contraseña que señala que la combinación de rango actual está asignada al mismo filtro que la combinación de rango previa; y asignen la combinación de rango actual a un filtro identificado por una segunda contraseña en respuesta a la recepción de la segunda contraseña que señala que la combinación de rango actual está asignada a un filtro diferente que la combinación de rango previa.

[0013] En otro ejemplo, un procedimiento de codificación de datos de vídeo incluye determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa; y generar una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, en el que la

segunda contraseña identifica un filtro asignado a la combinación de rango actual.

[0014] En otro ejemplo, un aparato para codificar datos de vídeo incluye un codificador de vídeo configurado para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa; y generar una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, en el que la segunda contraseña identifica un filtro asignado a la combinación de rango actual.

[0015] En otro ejemplo, un aparato incluye medios para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; medios para generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa; y medios para generar una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, en el que la segunda contraseña identifica un filtro asignado a la combinación de rango actual.

[0016] En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que cuando se ejecutan hacen que uno o más procesadores determinen una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; generen una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa; y generen una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, en el que la segunda contraseña identifica un filtro asignado a la combinación de rango actual.

[0017] Los detalles de uno o más ejemplos se exponen en las figuras adjuntas y la descripción a continuación. Otros rasgos característicos, objetivos y ventajas resultarán evidentes a partir de la descripción y las figuras, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LAS FIGURAS

[0018]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo ejemplar.

Las FIGS. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de división de árbol cuaternario aplicada a una unidad de codificación más grande (LCU).

Las FIGS. 2C y 2D son diagramas conceptuales que ilustran un ejemplo de un mapa de filtro para una serie de bloques de vídeo correspondientes la división de árbol cuaternario de ejemplo de las FIGS. 2A y 2B.

La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar consecuente con la presente divulgación.

La FIG. 4A es un diagrama conceptual que ilustra una asignación de rangos para dos métricas a filtros.

La FIG. 4B es un diagrama conceptual que ilustra una asignación de rangos para una métrica de actividad y una métrica de dirección a filtros.

La FIG. 5 es un diagrama de bloques que ilustra un decodificador de vídeo ejemplar consecuente con la presente divulgación.

Las FIGS. 6A, 6B y 6C muestran diagramas conceptuales de un bloque de píxeles 4x4.

La FIG. 7 es un diagrama de flujo que ilustra técnicas de codificación consecuentes con la presente divulgación.

Las FIGS. 8A y 8B son diagramas de flujo que ilustran técnicas de codificación consecuentes con la presente divulgación.

Las FIGS. 9A y 9B son diagramas de flujo que ilustran técnicas de codificación consecuentes con la presente divulgación.

La FIG. 10 es un diagrama de flujo que ilustra técnicas de codificación consecuentes con la presente divulgación.

La FIG. 11 es un diagrama de flujo que ilustra técnicas de codificación consecuentes con la presente divulgación.

DESCRIPCIÓN DETALLADA

[0019] La presente divulgación describe técnicas asociadas al filtrado de datos de vídeo en un proceso de codificación de vídeo y/o decodificación de vídeo. De acuerdo con la presente divulgación, el filtrado se aplica en un codificador, y la información del filtro se codifica en el flujo de bits para posibilitar que un decodificador identifique el filtrado que se aplicó en el codificador. El decodificador recibe datos de vídeo codificados que incluyen la información del filtro, descodifica los datos de vídeo y aplica el filtrado en base a la información de filtrado. De esta manera, el decodificador aplica el mismo filtrado que se aplicó en el codificador. De acuerdo con las técnicas de la presente divulgación, de trama en trama, de fragmento en fragmento o de LCU en LCU, un codificador puede seleccionar uno o más conjuntos de filtros, y de unidad codificada en unidad codificada, el codificador puede determinar si se aplica o no el filtrado. Para las unidades codificadas (CU) que se van a filtrar, el codificador puede realizar el filtrado de píxel en píxel o de grupo en grupo, donde un grupo, por ejemplo, podría ser un bloque de píxeles 2x2 o un bloque de píxeles 4x4.

[0020] De acuerdo con las técnicas de la presente divulgación, los datos de vídeo se pueden codificar en unidades denominadas unidades codificadas (CU). Las CU se pueden dividir en CU más pequeñas, o subunidades, usando un esquema de división de árbol cuaternario. La sintaxis que identifica el esquema de división de árbol cuaternario para una CU particular se puede transmitir desde un codificador a un decodificador. Se pueden filtrar múltiples entradas asociadas con cada subunidad de una CU dada durante el proceso de decodificación y reconstrucción de los datos de vídeo codificados. De acuerdo con las técnicas de la presente divulgación, la sintaxis de descripción de filtro puede describir un conjunto de filtros, tal como cuántos filtros hay en el conjunto o qué forma adoptan los filtros. La sintaxis adicional en el flujo de bits recibido por el decodificador puede identificar los filtros (es decir, los coeficientes de filtro) usados en el codificador para una subunidad particular. El filtro usado para una entrada particular se puede seleccionar en base a dos o métricas, donde determinadas combinaciones de valores para las dos o métricas se indexan a filtros específicos dentro de un conjunto de filtros. En otros casos, se pueden combinar dos o más métricas para formar una única métrica. La asignación de filtros a métricas también se puede señalar en el flujo de bits

[0021] Se pueden aplicar diferentes tipos de filtrado a píxeles o bloques de píxeles en base a dos o más métricas determinadas para los datos de vídeo. El filtro usado para un píxel particular se puede seleccionar en base a dos o más métricas, tales como alguna combinación de una métrica de actividad y una métrica de dirección. Una métrica de actividad, por ejemplo, puede cuantificar la actividad asociada a uno o más bloques de píxeles dentro de los datos de vídeo. La métrica de actividad puede comprender una métrica de varianza indicativa de la varianza de píxel dentro de un conjunto de píxeles. Una métrica de actividad puede ser específica de la dirección o bien no específica de la dirección. Por ejemplo, una métrica de actividad no específica de la dirección puede incluir un valor laplaciano modificado por suma, como se explica con mayor detalle a continuación.

[0022] Los ejemplos de métricas de actividad específicas de la dirección incluyen una métrica de actividad horizontal, una métrica de actividad vertical, una métrica de actividad de 45 grados y una métrica de actividad de 135 grados. Una métrica de dirección para un bloque de píxeles puede cuantificar cualquiera de la actividad horizontal, actividad vertical o actividad diagonal de un píxel o grupo de píxeles, o una métrica de dirección puede incluir una comparación de la actividad horizontal, actividad vertical y/o actividad diagonal, donde la actividad horizontal se refiere en general a cambios en los valores de píxel en una dirección horizontal, la actividad vertical se refiere en general a cambios en los valores de píxel en una dirección vertical, y la actividad diagonal se refiere en general a cambios en los valores de píxel en una dirección diagonal.

[0023] De acuerdo con las técnicas de la presente divulgación, cuando se determina un filtro para un bloque de píxeles, se puede usar un subconjunto de píxeles dentro del bloque para reducir la complejidad de codificación y decodificación. Por ejemplo, cuando se determina un filtro para un bloque de píxeles 4x4, puede que no sea necesario usar los dieciséis píxeles del bloque 4x4. Adicionalmente, de acuerdo con las técnicas de la presente divulgación, el subconjunto de píxeles dentro de un bloque actual que se está codificando se puede seleccionar de modo que las métricas se calculen solo usando valores de píxel del bloque actual y no valores de píxel de bloques contiguos. Por ejemplo, se podría calcular la métrica para un píxel que se está evaluando en base a la comparación del píxel con los píxeles cercanos. En algunos casos, uno o más de los píxeles cercanos para el píxel que se está evaluando podría estar en un bloque diferente del píxel que se está evaluando. En otros casos, sin embargo, uno o más de los píxeles cercanos para el píxel podrían estar en el mismo bloque que el píxel. De acuerdo con las técnicas de la presente divulgación, el subconjunto de píxeles se puede seleccionar para incluir píxeles que no tengan píxeles cercanos en bloques contiguos. Adicionalmente o de forma alternativa, el subconjunto de píxeles puede incluir píxeles que tengan píxeles cercanos en bloques contiguos, pero esos píxeles cercanos en bloques contiguos no se pueden usar cuando se determina la métrica. Al basar la determinación de una métrica particular en píxeles dentro de un bloque actual y no en píxeles de bloques contiguos, la necesidad de memorias intermedias en el codificador y/o decodificador, en algunos casos, se puede reducir o incluso eliminar.

[0024] En algunos casos, de acuerdo con las técnicas de la presente divulgación, el subconjunto de píxeles dentro de un bloque actual que se está codificando se puede seleccionar de modo que las métricas se calculen

solo usando los valores de píxel del bloque actual y los bloques contiguos izquierdo y derecho, pero no los valores de píxel de bloques contiguos superiores o bloques contiguos inferiores. Como resultado del orden de exploración de trama usado cuando se codifican bloques de vídeo, las memorias intermedias de línea para los bloques contiguos superior e inferior tienden a necesitar almacenar muchos más valores de píxel que las memorias intermedias de línea para almacenar valores de píxel de los bloques contiguos izquierdo y derecho.

[0025] De acuerdo con las técnicas de la presente divulgación, una unidad de filtro, tal como un filtro en bucle adaptativo, se puede configurar para utilizar múltiples filtros en base a la asignación de filtro multimétrica. Los filtros múltiples se pueden usar junto con una única entrada o múltiples entradas. Como se describirá con más detalle a continuación, las múltiples entradas descritas en la presente divulgación se refieren en general a datos de bloques de vídeo intermedios o datos de imagen que se producen durante los procesos de codificación y decodificación. Las múltiples entradas asociadas a un bloque de vídeo dado pueden incluir, por ejemplo, un bloque o imagen reconstruido (RI), un bloque o imagen reconstruido predespixelado (pRI), un bloque o imagen de predicción (PI) y/o una imagen de error (EI) de predicción cuantificada. En un esquema de única entrada, un filtro solo se puede aplicar a una de las entradas anteriores, tal como RI. Además, como se explica con mayor detalle a continuación, se pueden aplicar las técnicas de filtrado de la presente divulgación a las CU de diversos tamaños usando un esquema de división de árbol cuaternario. Al utilizar múltiples filtros con asignación de filtro multimétrica para las CU divididas usando un esquema de división de árbol cuaternario, se podría mejorar el rendimiento de codificación de vídeo, medido por una o ambas de la tasa de compresión y la calidad de vídeo reconstruido.

[0026] Para implementar las técnicas de filtrado multimétrico descritas anteriormente, un codificador mantiene, al generar, actualizar, almacenar u otros medios, una asignación de combinaciones de rangos a filtros. Como un ejemplo, la combinación de un primer rango para una primera métrica y un primer rango para una segunda métrica se puede correlacionar con un primer filtro. La combinación del primer rango para la primera métrica y un segundo rango para la segunda métrica también se puede correlacionar con el primer filtro o se puede correlacionar con un segundo filtro. Si una primera métrica tiene ocho rangos y una segunda métrica tiene cuatro rangos, por ejemplo, entonces la primera y la segunda métrica pueden tener treinta y dos combinaciones de rangos, y cada una de las treinta y dos combinaciones se puede correlacionar con un filtro. Sin embargo, cada combinación no se correlaciona necesariamente con un filtro único. Por tanto, las treinta y dos combinaciones se podrían correlacionar con cuatro filtros, ocho filtros, diez filtros o algún otro número de filtros. Para aplicar los mismos filtros que un codificador, un decodificador también puede mantener las mismas correlaciones de combinaciones de rango con respecto a filtros.

[0027] La presente divulgación describe técnicas para señalar desde un codificador a un decodificador, en un flujo de bits codificado, una asignación de combinaciones de rango a filtros. La asignación, por ejemplo, puede asociar cada combinación de rango a una identificación (ID) de filtro. Una manera simple de señalar esta asignación es usar una contraseña para cada ID de filtro y a continuación, para cada combinación de rangos, enviar la contraseña de la ID de filtro correspondiente. Esta técnica, sin embargo, es típicamente ineficiente. Las técnicas de la presente divulgación pueden aprovechar correlaciones dentro de la asignación usando procedimientos de codificación diferencial. Las combinaciones de rangos que comparten un rango común a veces usan el mismo filtro. Como un ejemplo, la combinación de un primer rango para una primera métrica y un primer rango para una segunda métrica y la combinación del primer rango para la primera métrica y un segundo rango para la segunda métrica comparten un rango común (el primer rango de la primera métrica). Por tanto, estas dos combinaciones, en algunos casos, se podrían correlacionar con la misma ID de filtro. Al aprovechar esta correlación, las técnicas de la presente divulgación pueden reducir el número de bits necesarios para señalar la asignación de combinaciones de rango para filtrar las ID desde un codificador a un decodificador.

[0028] Además de señalar la asignación de combinaciones de rango a las ID de filtro, la presente divulgación también describe técnicas para señalar, en un flujo de bits codificado, coeficientes de filtro para los filtros. Las técnicas de la presente descripción incluyen usar procedimientos de codificación diferencial para señalar coeficientes de filtro desde un codificador a un decodificador. De esta manera, se podrían comunicar los coeficientes de filtro para un segundo filtro a un decodificador como información de diferencia, donde la información de diferencia describe cómo modificar los coeficientes de filtro de un primer filtro de una manera que produce los coeficientes de filtro del segundo filtro. Las técnicas de codificación diferencial pueden ser más eficaces (es decir, pueden dar como resultado un mayor ahorro de bits) cuando los coeficientes de filtro del primer y el segundo filtro son más similares en comparación con cuando los coeficientes de filtro del primer y el segundo filtro son menos similares. Las técnicas de la presente divulgación incluyen determinar un orden secuencial en el que señalar los coeficientes de filtro para los filtros. Los órdenes determinados usando las técnicas descritas en la presente divulgación pueden dar como resultado una codificación diferencial mejorada de los coeficientes de filtro y, por tanto, en algunos casos dan como resultado un ahorro de bits cuando señalan los coeficientes de filtro.

[0029] Aunque las técnicas de la presente divulgación a veces se pueden describir con referencia al filtrado en bucle, las técnicas se pueden aplicar al filtrado en bucle, filtrado posterior al bucle y otros esquemas de filtrado tales como el filtrado conmutado. El filtrado en bucle se refiere en general al filtrado en el que los datos filtrados son parte de los bucles de codificación y decodificación de modo que los datos filtrados se usan para la intra o intercodificación predictiva. El filtrado posterior al bucle se refiere al filtrado que se aplica a los datos de vídeo

reconstruidos después del bucle de codificación. Con el filtrado posterior al bucle, los datos no filtrados, a diferencia de los datos filtrados, se usan para la intra o intercodificación predictiva. En algunas implementaciones, el tipo de filtrado puede conmutar entre el filtrado posterior al bucle y el filtrado en bucle, por ejemplo, de trama en trama, de fragmento en fragmento u otra base de este tipo, y la decisión de usar el filtrado posterior al bucle o el filtrado en bucle se puede señalar desde el codificador al descodificador para cada trama, fragmento, etc. Las técnicas de la presente divulgación no se limitan al filtrado en bucle o filtrado posterior, y se pueden aplicar a una amplia gama de filtrado aplicado durante la codificación de vídeo.

[0030] En la presente divulgación, el término "codificación" se refiere a la codificación o la descodificación. De forma similar, el término "codificador" se refiere en general a cualquier codificador de vídeo, descodificador de vídeo o codificador/descodificador combinado (códec). En consecuencia, el término "codificador" se usa en el presente documento para referirse a un dispositivo o aparato informático especializado que realiza la codificación de vídeo o la descodificación de vídeo.

[0031] Adicionalmente, en la presente divulgación, el término "filtro" se refiere en general a un conjunto de coeficientes de filtro. Por ejemplo, un filtro 3x3 se puede definir por un conjunto de 9 coeficientes de filtro, un filtro 5x5 se puede definir por un conjunto de 25 coeficientes de filtro, un filtro 9x5 se puede definir por un conjunto de 45 coeficientes de filtro y así sucesivamente. El término "conjunto de filtros" se refiere en general 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. De acuerdo con las técnicas descritas en la presente divulgación, para una serie de bloques de vídeo, tales como una trama, fragmento o unidad de codificación más grande (LCU), la información que identifica los conjuntos de filtros se señala desde el codificador al descodificador en un encabezado para la serie de los bloques de vídeo. El término "forma", a veces denominado "soporte de filtro", se refiere en general al número de filas de coeficientes de filtro y al número de columnas de coeficientes de filtro para un filtro particular. Por ejemplo, 9x9 es un ejemplo de una primera forma, 9x5 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 rombo, formas similares a rombos, formas circulares, formas similares a círculos, formas hexagonales, formas octogonales, formas cruzadas, formas de X, formas de T, otras formas geométricas u otras numerosas formas o configuración.

[0032] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo 110 ejemplar que puede implementar las técnicas de la presente divulgación. Como se muestra en la FIG. 1, el sistema 110 incluye un dispositivo de origen 112 que transmite datos de vídeo codificado a un dispositivo de destino 116 por medio de un canal de comunicación 115. El dispositivo de origen 112 y el dispositivo de destino 116 pueden comprender cualquiera de una amplia gama de dispositivos. En algunos casos, el dispositivo de origen 112 y el dispositivo de destino 116 pueden comprender auriculares de dispositivos de comunicación inalámbrica, tales como los llamados radioteléfonos móviles o satelitales. Sin embargo, las técnicas de la presente divulgación que se aplican de más en general al filtrado de datos de vídeo, no se limitan necesariamente a aplicaciones o configuraciones inalámbricas, y se pueden aplicar a dispositivos no inalámbricos, incluyendo las capacidades de codificación y/o descodificación de vídeo.

[0033] En el ejemplo de la FIG. 1, el dispositivo de origen 112 incluye una fuente de vídeo 120, un codificador de vídeo 122, un modulador/desmodulador (módem) 123 y un transmisor 124. El dispositivo de destino 116 incluye un receptor 126, un módem 127, un descodificador de vídeo 128 y un dispositivo de visualización 130. De acuerdo con la presente divulgación, el codificador de vídeo 122 del dispositivo de origen 112 se puede configurar para seleccionar uno o más conjuntos de coeficientes de filtro para múltiples entradas en un proceso de filtrado de bloques de vídeo y a continuación codificar el uno o más conjuntos seleccionados de coeficientes de filtro. Se pueden seleccionar filtros específicos del uno o más conjuntos de coeficientes de filtro en base a una o más métricas para una o más entradas, y los coeficientes de filtro se pueden usar para filtrar la una o más entradas. Las técnicas de filtrado de la presente divulgación en general son compatibles con cualquier técnica para codificar o señalar coeficientes de filtro en un flujo de bits codificado.

[0034] De acuerdo con las técnicas de la presente divulgación, un dispositivo que incluye el codificador de vídeo 122 puede señalar a un dispositivo que incluye el descodificador de vídeo 128 uno o más conjuntos de coeficientes de filtro para una serie de bloques de vídeo, tales como una trama o un fragmento. Para la serie de bloques de vídeo, el codificador de vídeo 122, por ejemplo, puede señalar un conjunto de filtros que se van a usar con todas las entradas, o puede señalar múltiples conjuntos de filtros que se van a usar con múltiples entradas (un conjunto por entrada, por ejemplo). Cada bloque de vídeo o CU dentro de la serie de bloques de vídeo puede contener a continuación una sintaxis adicional para identificar qué filtro o filtros del conjunto de filtros se va(n) a usar para cada entrada de ese bloque de vídeo, o de acuerdo con las técnicas de la presente divulgación, qué filtro o filtros del conjunto de filtros que se va(n) a usar se puede(n) determinar en base a dos o más métricas asociadas a una o más de las entradas.

[0035] Más específicamente, el codificador de vídeo 122 del dispositivo de origen 112 puede seleccionar uno o más conjuntos de filtros para una serie de bloques de vídeo, aplicar filtros del/de los conjunto(s) a píxeles o grupos de píxeles de entradas asociadas con las CU de la serie de bloques de vídeo durante el proceso de codificación,

y a continuación codificar los conjuntos de filtros (es decir, conjuntos de coeficientes de filtro) para su comunicación al descodificador de vídeo 128 del dispositivo de destino 116. El codificador de vídeo 122 puede determinar una o más métricas asociadas a las entradas de las CU codificadas para seleccionar qué filtro(s) del/de los conjunto(s) de filtros usar con píxeles o grupos de píxeles para esa CU particular. El codificador de vídeo 122 también puede señalar al descodificador de vídeo 128, como parte del flujo de bits codificado, una asignación de combinaciones de rangos a filtros dentro de un conjunto de filtros.

[0036] En el lado del descodificador, el descodificador de vídeo 128 puede determinar los coeficientes de filtro en base a la información del filtro recibida en la sintaxis de flujo de bits. El descodificador de vídeo 128 puede descodificar los coeficientes de filtro en base a descodificación directa o descodificación predictiva dependiendo de cómo se codificaron los coeficientes de filtro, que se pueden señalar como parte de la sintaxis de flujo de bits. Adicionalmente, el flujo de bits puede incluir información de sintaxis de descripción de filtro para describir los filtros para un conjunto de filtros. En base a la sintaxis de descripción de filtro, el descodificador 128 puede reconstruir los coeficientes de filtro en base a la información adicional recibida desde el codificador 122. El sistema 110 ilustrado de la FIG. 1 es simplemente ejemplar. Las técnicas de filtrado de la presente divulgación se pueden realizar por cualquier dispositivo de codificación o descodificación. El dispositivo de origen 112 y el dispositivo de destino 116 son simplemente ejemplos de dispositivos de codificación que pueden admitir dichas técnicas. El descodificador de vídeo 128 también puede determinar la asignación de combinaciones de rangos a filtros en base a la información del filtro recibida en la sintaxis de flujo de bits.

[0037] El codificador de vídeo 122 del dispositivo de origen 112 puede codificar datos de vídeo recibidos desde la fuente de vídeo 120 usando las técnicas de la presente divulgación. La fuente de vídeo 120 puede comprender un dispositivo de captación de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo captado previamente o una transmisión de vídeo desde un proveedor de contenido de vídeo. Como otra alternativa, la fuente de vídeo 120 puede generar datos basados en gráficos de ordenador como fuente de vídeo, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 120 es una cámara de vídeo, el dispositivo de origen 112 y el dispositivo de destino 116 pueden formar los llamados teléfonos con cámara o videoteléfonos. En cada caso, el vídeo capado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 122.

[0038] Una vez que los datos de vídeo se codifican por el codificador de vídeo 122, la información de vídeo codificada se puede modular a continuación por el módem 123 de acuerdo con una norma de comunicación, por ejemplo, tal como el acceso múltiple por división de código (CDMA), el acceso múltiple por división de frecuencia (FDMA), el multiplexado por división ortogonal de frecuencia (OFDM) o cualquier otra norma o técnica de comunicación, y transmitir al dispositivo de destino 116 por medio del transmisor 124. El módem 123 puede incluir diversos mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. El transmisor 124 puede incluir circuitos diseñados para transmitir datos, incluyendo amplificadores, filtros y una o más antenas.

[0039] El receptor 126 del dispositivo de destino 116 recibe información sobre el canal 115, y el módem 127 desmodula la información. El proceso de descodificación de vídeo realizado por el descodificador de vídeo 128 puede incluir filtrado, por ejemplo, como parte de la descodificación en bucle o como una etapa de filtrado posterior que sigue al bucle de descodificación. De cualquier manera, el conjunto de filtros aplicados por el descodificador de vídeo 128 para un fragmento o trama particular se puede descodificar usando las técnicas de la presente divulgación. La información del filtro descodificada puede incluir identificar la sintaxis de descripción de filtro en el flujo de bits codificado. Si, por ejemplo, se usa la codificación predictiva para los coeficientes de filtro, se pueden aprovechar las similitudes entre diferentes coeficientes de filtro para reducir la cantidad de información transmitida por el canal 115. En particular, un filtro (es decir, un conjunto de coeficientes de filtro) se puede codificar de forma predictiva como valores de diferencia en relación con otro conjunto de coeficientes de filtro asociados a un filtro diferente. El filtro diferente, por ejemplo, se puede asociar con un fragmento o trama diferente. En dicho caso, el descodificador de vídeo 128 podría recibir un flujo de bits codificado que comprende bloques de vídeo y filtrar información que identifica la trama o fragmento diferente con el que el filtro diferente es el filtro asociado. La información del filtro también incluye valores de diferencia que definen el filtro actual en relación con el filtro de la CU diferente. En particular, los valores de diferencia pueden comprender valores de diferencia de coeficiente de filtro que definen coeficientes de filtro para el filtro actual en relación con los coeficientes de filtro de un filtro diferente usado para una CU diferente.

[0040] El descodificador de vídeo 128 descodifica los bloques de vídeo, genera los coeficientes de filtro y filtra los bloques de vídeo descodificados en base a los coeficientes de filtro generados. El descodificador de vídeo 128 puede generar los coeficientes de filtro en base a la sintaxis de descripción de filtro recuperada del flujo de bits. Los bloques de vídeo descodificados y filtrados se pueden ensamblar en tramas de vídeo para formar datos de vídeo descodificados. El dispositivo de visualización 128 visualiza los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0041] El canal de comunicación 115 puede comprender cualquier medio de comunicación inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 115 puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El canal de comunicación 115 representa en general cualquier medio de comunicación adecuado o una colección de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo de origen 112 al dispositivo de destino 116. De nuevo, la FIG. 1 es simplemente ejemplar y las técnicas de la presente divulgación se pueden aplicar a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyan necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, se podrían recuperar los datos de una memoria local, transmitirse sobre una red o similar.

[0042] De forma alternativa, se pueden emitir los datos codificados desde el codificador de vídeo 122 a un dispositivo de almacenamiento 132. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 132 por el descodificador de vídeo 128. El dispositivo de almacenamiento 132 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso local o distribuidos, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 132 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 112. El dispositivo de destino 116 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 132 por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 116. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para una página web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 132 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0043] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 110 se puede configurar para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0044] El codificador de vídeo 122 y el descodificador de vídeo 128 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma H.264 de UIT-T, denominada de forma alternativa MPEG-4, parte 10, codificación de vídeo avanzada (AVC), que se usará en partes de la presente divulgación con propósitos de explicación. Sin embargo, muchas de las técnicas de la presente divulgación se pueden aplicar fácilmente a cualquiera de una variedad de otras normas de codificación de vídeo, incluyendo la norma HEVC surgida recientemente. En general, cualquier norma que permite el filtrado en el codificador y descodificador se puede beneficiar de diversos aspectos de la enseñanza de la presente divulgación.

[0045] Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 122 y el descodificador de vídeo 128 se pueden integrar cada uno con un codificador y un descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si fuera aplicable, las unidades MUX-DEMUX pueden estar conformes con el protocolo multiplexor H.223 de UIT u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0046] El codificador de vídeo 122 y el descodificador de vídeo 128 se pueden implementar cada uno como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno del codificador de vídeo 122 y el descodificador de vídeo 128 se pueden incluir en uno o más codificadores o descodificadores, pudiendo cualquiera de los mismos estar integrado como parte de un codificador/descodificador combinado (CÓDEC) en un dispositivo móvil, dispositivo de abonado, dispositivo de radiodifusión, servidor respectivo o similares.

[0047] En algunos casos, los dispositivos 112, 116 pueden funcionar de una manera sustancialmente simétrica.

Por ejemplo, cada uno de los dispositivos 112, 116 puede incluir componentes de codificación y decodificación de vídeo. Por consiguiente, el sistema 110 pueda admitir la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 112, 116, por ejemplo, para transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

[0048] Durante el proceso de codificación, el codificador de vídeo 122 puede ejecutar una serie de técnicas o etapas de codificación. En general, un codificador de vídeo 122 funciona sobre bloques de vídeo dentro de tramas de vídeo individuales para codificar los datos de vídeo. En un ejemplo, un bloque de vídeo puede corresponder a un macrobloque o una división de un macrobloque. Los macrobloques son un tipo de bloque de vídeo definido por la norma H.264 de UIT y otras normas. Los macrobloques típicamente se refieren a bloques de datos de 16x16, aunque el término también se usa a veces para referirse genéricamente a cualquier bloque de vídeo de tamaño NxN o NxM. La norma H.264 de UIT-T admite la intrapredicción en diversos tamaños de bloque, tales como 16x16, 8x8 o 4x4 para componentes de luma y 8x8 para componentes de croma, así como la interpredicción en diversos tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y tamaños ajustados a escala correspondientes para componentes de croma. En la presente divulgación, "NxN" se refiere a las dimensiones de píxel del bloque en lo que respecta a las dimensiones vertical y horizontal, por ejemplo 16x16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical y 16 píxeles en una dirección horizontal. Asimismo, un bloque de NxN tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero positivo. Los píxeles de un bloque se pueden disponer en filas y columnas.

[0049] La norma HEVC de reciente surgimiento define nuevos términos para bloques de vídeo. En particular, los bloques de vídeo (o divisiones de los mismos) se pueden denominar "unidades de codificación" (o CU). Con la norma HEVC, las unidades codificadas más grandes (LCU) se pueden dividir en CU más pequeñas de acuerdo con un esquema de división de árbol cuaternario, y las diferentes CU que se definen en el esquema se pueden dividir además en las llamadas unidades de predicción (PU). Las LCU, las CU y las PU son todas bloques de vídeo en el sentido de la presente divulgación. También se pueden usar otros tipos de bloques de vídeo, consecuentes con la norma HEVC u otras normas de codificación de vídeo. Por tanto, la frase "bloques de vídeo" se refiere a cualquier tamaño de bloque de vídeo. Se pueden incluir CU separadas para componentes de luma y tamaños ajustados a escala para componentes de croma para un píxel dado, aunque también se podrían usar otros espacios de color.

[0050] Los bloques de vídeo pueden tener tamaños fijos o variados y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Cada trama de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de bloques de vídeo, que se pueden disponer en divisiones, también denominadas subbloques. De acuerdo con el esquema de división de árbol cuaternario mencionado anteriormente y descrito con más detalle a continuación, una primera CU de N/2xN/2 puede comprender un subbloque de una LCU de NxN, una segunda CU de N/4xN/4 también puede comprender un subbloque de la primera CU. Una PU de N/8xN/8 puede comprender un subbloque de la segunda CU. De forma similar, como otro ejemplo, los tamaños de bloque que son menores de 16x16 se pueden denominar divisiones de un bloque de vídeo de 16x16 o subbloques del bloque de vídeo de 16x16. Asimismo, para un bloque de NxN, los tamaños de bloque menores a NxN se pueden denominar divisiones o subbloques del bloque de NxN. Los bloques de vídeo pueden comprender bloques de datos de píxeles en el dominio de píxel, o bloques de coeficientes de transformada en el dominio de transformada, por ejemplo, tras la aplicación de una transformada, tal como una transformada discreta del coseno (DCT), una transformada de números enteros, una transformada de ondículas o una transformada conceptualmente similar a los datos de bloques de vídeo residuales que representan diferencias de píxeles entre los bloques de vídeo codificados y los bloques de vídeo predictivos. En algunos casos, un bloque de vídeo puede comprender bloques de coeficientes de transformada cuantificados en el dominio de transformada.

[0051] Los datos de sintaxis dentro de un flujo de bits pueden definir una LCU para una trama o un fragmento, que es una unidad de codificación más grande en términos del número de píxeles para esa trama o fragmento. En general, una LCU o CU tiene un propósito similar a un macrobloque codificado de acuerdo con H.264, excepto que las LCU y las CU no tienen una distinción de tamaño específica. En cambio, un tamaño de LCU se puede definir de trama en trama o de fragmento en fragmento, y una LCU se puede separar en CU. En general, las referencias en la presente divulgación a una CU se pueden referir a una LCU de una imagen o a una sub-CU de una LCU. Una LCU se puede separar en sub-CU, y cada sub-CU se puede separar en sub-CU. Los datos de sintaxis para un flujo de bits pueden definir un número máximo de veces que se puede separar una LCU, denominado profundidad de CU. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). La presente divulgación también usa los términos "bloque" y "bloque de vídeo" para referirse a cualquiera de una LCU, CU, PU, SCU o TU.

[0052] Como se presenta anteriormente, una LCU se puede asociar a una estructura de datos de árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, donde un nodo raíz corresponde a la LCU. Si una CU se separa en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, de los que cada uno corresponde a una de las sub-CU. Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo en el árbol

cuaternario puede incluir un indicador de separación, que indica si la CU correspondiente al nodo está separada en sub-CU. Los elementos de sintaxis para una CU se pueden definir de manera recursiva y pueden depender de si la CU está separada en sub-CU.

[0053] Una CU que no está separada puede incluir una o más unidades de predicción (PU). En general, una PU representa la totalidad o una parte de la CU correspondiente e incluye datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU se codifica en intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica en intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una trama de referencia a la que apunta el vector de movimiento y/o una lista de referencia (por ejemplo, lista 0 o lista 1) para el vector de movimiento. Los datos para la CU que definen la(s) PU también pueden describir, por ejemplo, una división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está sin codificar, codificada en modo de intrapredicción o codificada en modo de interpredicción.

[0054] Una CU que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Las TU comprenden la estructura de datos que incluye coeficientes de transformada residuales, que típicamente se cuantifican. En particular, tras la predicción usando una PU, un codificador de vídeo puede calcular valores residuales para la parte de la CU correspondiente a la PU. Los valores residuales se pueden transformar, cuantificar, explorar y almacenar en una TU, que puede tener tamaños variables correspondientes al tamaño de la transformada que se realizó. En consecuencia, una TU no está necesariamente limitada al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que las PU correspondientes para la misma CU. En algunos ejemplos, el tamaño máximo de una TU puede ser el tamaño de la CU correspondiente. De nuevo, las TU pueden comprender las estructuras de datos que incluyen los coeficientes de transformada residuales asociados a una CU dada.

[0055] Las FIGS. 2A y 2B son diagramas conceptuales que ilustran un árbol cuaternario 250 de ejemplo y una LCU 272 correspondiente. La FIG. 2A representa un árbol cuaternario 250 de ejemplo, que incluye nodos dispuestos de forma jerárquica. Cada nodo en un árbol cuaternario, tal como el árbol cuaternario 250, puede ser un nodo hoja sin hijos, o tener cuatro nodos hijos. En el ejemplo de la FIG. 2A, el árbol cuaternario 250 incluye el nodo raíz 252. El nodo raíz 252 tiene cuatro nodos hijos, incluyendo los nodos hoja 256A-256C (nodos hoja 256) y el nodo 254. Como el nodo 254 no es un nodo hoja, el nodo 254 incluye cuatro nodos hijos, que en este ejemplo son los nodos hoja 258A-258D (nodos hoja 258).

[0056] El árbol cuaternario 250 puede incluir datos que describen características de una LCU correspondiente, tal como la LCU 272 en este ejemplo. Por ejemplo, el árbol cuaternario 250, por su estructura, puede describir la separación de la LCU en sub-CU. Supóngase que la LCU 272 tiene un tamaño $2N \times 2N$. La LCU 272, en este ejemplo, tiene cuatro sub-CU 276A-276C (sub-CU 276) y 274, cada una de tamaño $N \times N$. La sub-CU 274 se separa además en cuatro sub-CU 278A-278D (sub-CU 278), cada una de tamaño $N/2 \times N/2$. La estructura de árbol cuaternario 250 corresponde a la separación de la LCU 272, en este ejemplo. Es decir, el nodo raíz 252 corresponde a la LCU 272, los nodos hoja 256 corresponden a las sub-CU 276, el nodo 254 corresponde a las sub-CU 274 y los nodos hoja 258 corresponden a la sub-CU 278.

[0057] Los datos para los nodos del árbol cuaternario 250 pueden describir si la CU correspondiente al nodo está separada. Si la CU está separada, pueden estar presentes cuatro nodos adicionales en el árbol cuaternario 250. En algunos ejemplos, un nodo de un árbol cuaternario se puede implementar de manera similar al siguiente pseudocódigo:

```
quadtree _node {
    boolean split_flag (1);
    // signaling data
    if (split_flag) {
        quadtree node child1;
        quadtree node child2;
        quadtree node child3;
        quadtree node child4;
    }
}
```

El valor del indicador de separación puede ser un valor de un bit representativo de si la CU correspondiente al nodo actual está separada. Si la CU no está separada, el valor del indicador de separación puede ser '0', mientras que si la CU está separada, el valor de split_flag (indicador de separación) puede ser '1'. Con respecto al ejemplo del árbol cuaternario 250, una matriz de valores de indicador de separación puede ser 101000000.

[0058] En algunos ejemplos, cada una de las sub-CU 276 y las sub-CU 278 se puede codificar con intrapredicción usando el mismo modo de intrapredicción. En consecuencia, el codificador de vídeo 122 puede proporcionar una indicación del modo de intrapredicción en el nodo raíz 252. Además, determinados tamaños de sub-CU pueden tener múltiples transformadas posibles para un modo de intrapredicción particular. El codificador de vídeo 122 puede proporcionar una indicación de la transformada a usar para dichas sub-CU en el nodo raíz 252. Por ejemplo, las sub-CU de tamaño $N/2 \times N/2$ pueden tener múltiples transformadas posibles disponibles. El codificador de vídeo 122 puede señalar la transformada a usar en el nodo raíz 252. En consecuencia, el decodificador de vídeo 128 puede determinar la transformada a aplicar a las sub-CU 278 en base al modo de intrapredicción señalado en el nodo raíz 252 y la transformada señalada en el nodo raíz 252.

[0059] Como tal, el codificador de vídeo 122 no necesita transformadas de señal para aplicar a las sub-CU 276 y las sub-CU 278 en los nodos hoja 256 y los nodos hoja 258, sino que simplemente puede señalar un modo de intrapredicción y, en algunos ejemplos, una transformada a aplicar a determinados tamaños de sub-CU, en el nodo raíz 252, de acuerdo con las técnicas de la presente divulgación. De esta manera, estas técnicas pueden reducir el coste general de señalización de las funciones de transformada para cada sub-CU de una LCU, tal como la LCU 272.

[0060] En algunos ejemplos, los modos de intrapredicción para las sub-CU 276 y/o las sub-CU 278 pueden ser diferentes a los modos de intrapredicción para la LCU 272. El codificador de vídeo 122 y el decodificador de vídeo 130 se pueden configurar con funciones que asignan un modo de intrapredicción señalado en el nodo raíz 252 a un modo de intrapredicción disponible para las sub-CU 276 y/o las sub-CU 278. La función puede proporcionar una asignación de varios a uno de los modos de intrapredicción disponibles para la LCU 272 a los modos de intrapredicción para las sub-CU 276 y/o las sub-CU 278.

[0061] Un fragmento se puede dividir en bloques de vídeo (o LCU) y cada bloque de vídeo se puede dividir de acuerdo con la estructura de árbol cuaternario descrita en relación con las FIGS. 2A-B. Adicionalmente, como se muestra en la FIG. 2C, los subbloques de árbol cuaternario indicados por "ACTIVADO" (ON) se pueden filtrar por los filtros de bucle descritos en el presente documento, mientras que los subbloques de árbol cuaternario indicados por "DESACTIVADO" (OFF) no se pueden filtrar. La decisión de si filtrar o no un bloque o subbloque dado se puede determinar en el codificador comparando el resultado filtrado y el resultado no filtrado en relación con el bloque original que se está codificando. La FIG. 2D es un árbol de decisión que representa decisiones de división que dan como resultado la división de árbol cuaternario que se muestra en la FIG. 2C. El filtrado real aplicado a cualquier píxel para bloques "ACTIVADOS", se puede determinar en base a las métricas analizadas en el presente documento.

[0062] En particular, la FIG. 2C puede representar un bloque de vídeo relativamente grande que se divide de acuerdo con un esquema de división de árbol cuaternario en bloques de vídeo más pequeños de tamaños variados. Cada bloque de vídeo está marcado (activado (ON) o desactivado (OFF)) en la FIG. 2C, para ilustrar si se debe aplicar o evitar el filtrado para ese bloque de vídeo. El codificador de vídeo puede definir este mapa de filtro comparando versiones filtradas y no filtradas de cada bloque de vídeo con el bloque de vídeo original que se está codificando.

[0063] De nuevo, la FIG. 2D es un árbol de decisión correspondiente a las decisiones de división que dan como resultado la división de árbol cuaternario mostrada en la FIG. 2C. En la FIG. 2D, cada círculo puede corresponder a una CU. Si el círculo incluye un indicador "1", a continuación esa CU se divide además en cuatro CU más, pero si el círculo incluye un indicador "0", esa CU no se divide más. Cada círculo (por ejemplo, correspondiente a las CU) también incluye un rombo asociado. Si el indicador en el rombo para una CU dada se establece en 1, entonces se "activará" el filtrado para esa CU, pero si el indicador en el rombo para una CU dada se establece en 0, entonces se desactivará el filtrado. De esta manera, las FIGS. 2C y 2D se pueden ver individual o conjuntamente como un mapa de filtro que se puede generar en un codificador y comunicar a un decodificador al menos una vez por fragmento de datos de vídeo codificados para comunicar el nivel de división de árbol cuaternario para un bloque de vídeo dado (por ejemplo, una LCU) si aplicar o no el filtrado a cada bloque de vídeo dividido (por ejemplo, cada CU dentro de la LCU).

[0064] Los bloques de vídeo más pequeños pueden proporcionar una mejor resolución y se pueden usar para localizaciones de una trama de vídeo que incluyen altos niveles de detalle. Los bloques de vídeo más grandes pueden proporcionar una mayor eficacia de codificación y se pueden usar para localizaciones de una trama de vídeo que incluye un bajo nivel de detalle. Un fragmento se puede considerar una pluralidad de bloques y/o subbloques de vídeo. Cada fragmento puede ser una serie independientemente decodificable de bloques de vídeo de una trama de vídeo. De forma alternativa, las propias tramas pueden ser series decodificables de bloques de vídeo, u otras partes de una trama se pueden definir como series decodificables de bloques de vídeo. El término "series de bloques de vídeo" se puede referir a cualquier parte independientemente decodificable de una trama de vídeo, tal como una trama completa, un fragmento de una trama, un grupo de imágenes (GOP), denominado también secuencia, u otra unidad independientemente decodificable definida de acuerdo con las técnicas de codificación aplicables. Los aspectos de la presente divulgación se pueden describir en referencia a tramas o fragmentos, pero dichas referencias son simplemente ejemplares. Se debe entender que en general se puede usar

cualquier serie de bloques de vídeo en lugar de una trama o un fragmento.

[0065] Los datos de sintaxis se pueden definir por unidad codificada de modo que cada CU incluya datos de sintaxis asociados. La información del filtro descrita en el presente documento puede ser parte de dicha sintaxis para una CU, pero es más probable que sea parte de la sintaxis para una serie de bloques de vídeo, tal como una trama, un fragmento, un GOP, LCU o una secuencia de tramas de vídeo, en lugar de para una CU. Los datos de sintaxis pueden indicar el conjunto o conjuntos de filtros que se van a usar con las CU del fragmento o trama. Adicionalmente, no toda la información del filtro necesariamente se tiene que incluir en el encabezado de una serie común de bloques de vídeo. Por ejemplo, la sintaxis de descripción de filtro se podría transmitir en un encabezado de trama, mientras que otra información del filtro se señala en un encabezado para una LCU.

[0066] El codificador de vídeo 122 puede realizar una codificación predictiva en la que un bloque de vídeo que se codifica se compara con una trama predictiva (u otra CU) para identificar un bloque predictivo. Las diferencias entre el bloque de vídeo actual que se está codificando y el bloque predictivo se codifican como un bloque residual, y la sintaxis de predicción se usa para identificar el bloque predictivo. El bloque residual se puede transformar y cuantificar. Las técnicas de transformada pueden comprender un proceso DCT o un proceso conceptualmente similar, transformadas de enteros, transformadas de ondículas u otros tipos de transformadas. En un proceso DCT, como ejemplo, el proceso de transformada convierte un conjunto de valores de píxel en coeficientes de transformada, que pueden representar la energía de los valores de píxel en el dominio de frecuencia. La cuantificación típicamente se aplica a los coeficientes de transformada, y en general implica un proceso que limita el número de bits asociados a cualquier coeficiente de transformada dado.

[0067] Tras la transformada y la cuantificación, la codificación por entropía se puede realizar en los bloques de vídeo residuales cuantificados y transformados. Los elementos de sintaxis, tal como la información del filtro y los vectores de predicción definidos durante la codificación, también se pueden incluir en el flujo de bits codificado por entropía para cada CU. En general, la codificación por entropía comprende uno o más procesos que comprimen conjuntamente una secuencia de coeficientes de transformada cuantificados y/u otra información de sintaxis. Las técnicas de exploración, tales como las técnicas de exploración en zigzag, se realizan en los coeficientes de transformada cuantificados, por ejemplo, como parte del proceso de codificación por entropía, para definir uno o más vectores unidimensionales seriados de coeficientes a partir de bloques de vídeo bidimensionales. También se pueden usar otras técnicas de exploración, incluyendo otros órdenes de exploración o exploraciones adaptativas, y posiblemente señaladas en el flujo de bits codificado. En cualquier caso, los coeficientes explorados a continuación se codifican por entropía junto con cualquier información de sintaxis, por ejemplo, por medio de la codificación de longitud variable adaptativa al contenido (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC) u otro proceso de codificación por entropía.

[0068] Como parte del proceso de codificación, los bloques de vídeo codificados se pueden descodificar para generar los datos de vídeo usados para la posterior codificación basada en la predicción de los bloques de vídeo posteriores. En esta fase, se puede realizar el filtrado para mejorar la calidad de vídeo y, por ejemplo, eliminar los artefactos de efecto pixelado del vídeo descodificado. Los datos filtrados se pueden usar para la predicción de otros bloques de vídeo, caso en el que el filtrado se denomina filtrado "en bucle". De forma alternativa, la predicción de otros bloques de vídeo se puede basar en datos sin filtrar, caso en el que el filtrado se denomina "filtrado posterior".

[0069] De trama en trama, de fragmento en fragmento o de LCU en LCU, el codificador de vídeo 122 puede seleccionar uno o más conjuntos de filtros, y de unidad codificada en unidad codificada, el codificador puede determinar si se aplica o no el filtrado. Para las CU que se van a filtrar, el codificador puede realizar el filtrado de píxel en píxel o de grupo en grupo, donde un grupo podría, por ejemplo, ser un bloque de píxeles 2x2 o un bloque de píxeles 4x4. Estas selecciones se pueden hacer de una manera que promueva la calidad de vídeo. Dichos conjuntos de filtros se pueden seleccionar de conjuntos predefinidos de filtros, o se pueden definir de forma adaptativa para promover la calidad de vídeo. Como ejemplo, el codificador de vídeo 122 puede seleccionar o definir varios conjuntos de filtros para una trama o fragmento dado de modo que se usen diferentes filtros para diferentes píxeles o grupos de píxeles de las CU de esa trama o fragmento. En particular, para cada entrada asociada con una CU, se pueden definir varios conjuntos de coeficientes de filtro, y las dos o más métricas asociadas a los píxeles de la CU se pueden usar para determinar qué filtro del conjunto de filtros usar con dichos píxeles o grupos de píxeles.

[0070] En algunos casos, el codificador de vídeo 122 puede aplicar varios conjuntos de coeficientes de filtro y seleccionar uno o más conjuntos que producen el vídeo de mejor calidad en términos de cantidad de distorsión entre un bloque codificado y un bloque original, y/o los más altos niveles de compresión. En cualquier caso, una vez seleccionado, el conjunto de coeficientes de filtro aplicados por el codificador de vídeo 122 para cada CU se puede codificar y comunicar al descodificador de vídeo 128 del dispositivo de destino 118 de modo que el descodificador de vídeo 128 pueda aplicar el mismo filtrado que se aplicó durante el proceso de codificación para cada CU dada.

[0071] Cuando se usan dos o más métricas para determinar qué filtro usar con una entrada particular para una

CU, la selección del filtro para esa CU particular no necesariamente necesita comunicarse al decodificador de vídeo 128. En cambio, el decodificador de vídeo 128 también puede calcular las dos o más métricas, y en base a la información del filtro proporcionada previamente por el codificador de vídeo 122, hacer coincidir la combinación de dos o más métricas con un filtro particular.

[0072] La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo 350 consecuente con la presente divulgación. El codificador de vídeo 350 puede corresponder al codificador de vídeo 122 del dispositivo 120 o a un codificador de vídeo de un dispositivo diferente. Como se muestra en la FIG. 3, el codificador de vídeo 350 incluye un módulo de predicción 332, los sumadores 348 y 351 y una memoria 334. El codificador de vídeo 350 también incluye una unidad de transformada 338 y una unidad de cuantificación 340, así como una unidad de cuantificación inversa 342 y una unidad de transformada inversa 344. El codificador de vídeo 350 también incluye un filtro de eliminación de efecto pixelado 347 y una unidad de filtro adaptativo 349. El codificador de vídeo 350 también incluye una unidad de codificación por entropía 346. La unidad de filtro 349 del codificador de vídeo 350 puede realizar operaciones de filtrado y también puede incluir una unidad de selección de filtro (FSU) 353 para identificar un filtro o conjunto de filtros deseable o preferente que se va a usar para la decodificación. La unidad de filtro 349 también puede generar información del filtro que identifique los filtros seleccionados de modo que los filtros seleccionados se puedan comunicar eficazmente como información del filtro a otro dispositivo que se va a usar durante una operación de decodificación.

[0073] Durante el proceso de codificación, el codificador de vídeo 350 recibe un bloque de vídeo, tal como una LCU, que se va a codificar, y el módulo de predicción 332 realiza técnicas de codificación predictiva en el bloque de vídeo. Usando el esquema de división de árbol cuaternario analizado anteriormente, el módulo de predicción 332 puede dividir el bloque de vídeo y realizar técnicas de codificación predictiva en las CU de diferentes tamaños. Para la intercodificación, el módulo de predicción 332 compara el bloque de vídeo que se va a codificar, incluyendo los subbloques del bloque de vídeo, con diversos bloques en una o más tramas o fragmentos de referencia de vídeo para definir un bloque predictivo. Para la intracodificación, el módulo de predicción 332 genera un bloque predictivo en base a datos contiguos dentro de la misma CU. El módulo de predicción 332 emite el bloque de predicción y el sumador 348 resta el bloque de predicción del bloque de vídeo que se está codificando para generar un bloque residual.

[0074] Para la intercodificación, el módulo de predicción 332 puede comprender las unidades de estimación de movimiento y compensación de movimiento que identifican a un vector de movimiento que apunta a un bloque de predicción y genera el bloque de predicción en base al vector de movimiento. Típicamente, la estimación de movimiento se considera el proceso de generar el vector de movimiento, que estima el movimiento. Por ejemplo, el vector de movimiento puede indicar el desplazamiento de un bloque predictivo dentro de una trama predictiva en relación con el bloque actual que se está codificando dentro de la trama actual. La compensación de movimiento se considera típicamente el proceso de extraer o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento. Para la intracodificación, el módulo de predicción 332 genera un bloque predictivo en base a datos contiguos dentro de la misma CU. Uno o más modos de intrapredicción pueden definir cómo se puede definir un bloque de intrapredicción.

[0075] Después de que el módulo de predicción 332 emite el bloque de predicción y el sumador 348 resta el bloque de predicción del bloque de vídeo que se está codificando para generar un bloque residual, la unidad de transformada 338 aplica una transformada al bloque residual. La transformada puede comprender una transformada de coseno discreta (DCT) o una transformada conceptualmente similar tal como la definida por una norma de codificación tal como la norma HEVC. También se podrían usar transformadas de ondículas, transformadas de números enteros, transformadas de subbandas u otros tipos de transformadas. En cualquier caso, la unidad de transformada 338 aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual de un dominio de píxel en un dominio de frecuencia.

[0076] La unidad de cuantificación 340 a continuación cuantifica los coeficientes de transformada residuales para reducir adicionalmente la velocidad de transmisión de bits. La unidad de cuantificación 340, por ejemplo, puede limitar el número de bits usados para codificar cada uno de los coeficientes. Después de la cuantificación, la unidad de codificación por entropía 346 explora el bloque de coeficientes cuantificados de una representación bidimensional a uno o más vectores unidimensionales seriados. El orden de exploración se puede preprogramar para que se produzca en un orden definido (tal como exploración en zigzag, exploración horizontal, exploración vertical, combinaciones u otro orden predefinido), o posiblemente adaptativo definido en base a estadísticas de codificación previas.

[0077] Tras este proceso de exploración, la unidad de codificación por entropía 346 codifica los coeficientes de transformada cuantificados (junto con cualquier elemento de sintaxis) de acuerdo con una metodología de codificación por entropía, tal como CAVLC o CABAC, para comprimir adicionalmente los datos. Los elementos de sintaxis incluidos en el flujo de bits codificado por entropía pueden incluir la sintaxis de predicción del módulo de predicción 332, tal como vectores de movimiento para intercodificación o modos de predicción para intracodificación. Los elementos de sintaxis incluidos en el flujo de bits codificado por entropía también pueden

incluir información del filtro de la unidad de filtro 349, que se puede codificar de la manera descrita en el presente documento.

[0078] CAVLC es un tipo de técnica de codificación por entropía admitida por la norma H.264/MPEG4 de UIT, AVC, que se puede aplicar de forma vectorializada por la unidad de codificación por entropía 346. CAVLC usa tablas de codificación de longitud variable (VLC) de una manera que comprime eficazmente las "ejecuciones" seriadas de coeficientes de transformada y/o elementos de sintaxis. CABAC es otro tipo de técnica de codificación por entropía admitida por la norma H.264/MPEG4 de UIT, AVC, que se puede aplicar de forma vectorializada por la unidad de codificación por entropía 346. CABAC implica varias fases, incluyendo binarización, selección del modelo de contexto y codificación aritmética binaria. En este caso, la unidad de codificación por entropía 346 codifica los coeficientes de transformada y los elementos de sintaxis de acuerdo con CABAC. Al igual que la norma H.264/MPEG4 de IUT, AVC, la norma HEVC de reciente surgimiento también puede admitir tanto la codificación por entropía CAVLC como CABAC. Además, también existen muchos otros tipos de técnicas de codificación por entropía, y en el futuro probablemente surgirán nuevas técnicas de codificación por entropía. La presente divulgación no se limita a ninguna técnica específica de codificación por entropía.

[0079] Tras la codificación por entropía realizada por la unidad de codificación por entropía 346, el vídeo codificado se puede transmitir a otro dispositivo o archivar para su transmisión o recuperación posterior. De nuevo, el vídeo codificado puede comprender los vectores codificados por entropía y diversas sintaxis, que se pueden usar por el descodificador para configurar apropiadamente el proceso de descodificación. La unidad de cuantificación inversa 342 y la unidad de transformada inversa 344 aplican la cuantificación inversa y la transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel. El sumador 351 añade el bloque residual reconstruido al bloque de predicción producido por el módulo de predicción 332 para producir un bloque de vídeo reconstruido predespixelado, a veces denominado imagen reconstruida predespixelada. El filtro de eliminación de efecto pixelado 347 puede aplicar filtrado al bloque de vídeo reconstruido predespixelado para mejorar la calidad de vídeo al eliminar el efecto pixelado u otros artefactos. La salida del filtro de eliminación de efecto pixelado 347 se puede denominar bloque de vídeo posdespixelado, bloque de vídeo reconstruido o imagen reconstruida.

[0080] La unidad de filtro 349 se puede configurar para recibir una única entrada o múltiples entradas. En el ejemplo de la FIG. 3, la unidad de filtro 349 recibe como entrada la imagen reconstruida (RI) posdespixelada, la imagen reconstruida predespixelada (pRI), la imagen de predicción (PI) y el bloque residual reconstruido (EI). La unidad de filtro 349 puede usar cualquiera de estas entradas individualmente o bien en combinación para producir una imagen reconstruida para almacenar en la memoria 334. Adicionalmente, como se analizará con más detalle a continuación, en base a dos o más métricas, se pueden seleccionar uno o más filtros que se van a aplicar a la(s) entrada(s). En un ejemplo, la salida de la unidad de filtro 349 puede ser un filtro adicional aplicado a RI. En otro ejemplo, la salida de la unidad de filtro 349 puede ser un filtro adicional aplicado a pRI. En otros ejemplos, sin embargo, la salida de la unidad de filtro 349 se puede basar en múltiples entradas. Por ejemplo, la unidad de filtro 349 puede aplicar un primer filtro a pRI y a continuación usar la versión filtrada de pRI junto con las versiones filtradas de EI y PI para crear una imagen reconstruida. En los casos donde la salida de la unidad de filtro 349 es el producto de un filtro adicional que se aplica a una única entrada, la unidad de filtro 349 de hecho puede aplicar filtros a las otras entradas, pero esos filtros pueden tener todos los coeficientes cero. De forma similar, si la salida de la unidad de filtro 349 es el producto de aplicar tres filtros a tres entradas, la unidad de filtro 349 de hecho puede aplicar un filtro a la cuarta entrada, pero ese filtro podría tener todos los coeficientes cero.

[0081] La unidad de filtro 349 también se puede configurar para recibir una única entrada. Por ejemplo, aunque la FIG. 3 muestra que PI, EI, pRI y RI se introducen en la unidad de filtro 349, en algunas implementaciones RI podría ser la única entrada recibida por la unidad de filtro 349. En una implementación de este tipo, la unidad de filtro 349 podría aplicar un filtro a RI de modo que una versión filtrada de RI sea más similar a la imagen original que la versión no filtrada de RI. En otras implementaciones, la unidad de filtro 349 y el filtro de eliminación de efecto pixelado 347 se pueden combinar en una única unidad de filtrado que aplica el filtrado a pRI. Las técnicas de la presente divulgación, que en general se refieren a la asignación de filtros basada en múltiples métricas, son compatibles tanto con esquemas de filtrado de única entrada como de múltiples entradas que utilizan múltiples filtros.

[0082] El filtrado por la unidad de filtro 349 puede mejorar la compresión al generar bloques de vídeo predictivos que coinciden más estrechamente con los bloques de vídeo que se están codificando que los bloques de vídeo predictivos no filtrados. Después del filtrado, el bloque de vídeo reconstruido se puede usar por el módulo de predicción 332 como un bloque de referencia para intercodificar un bloque en una trama de vídeo posterior u otra CU. Aunque la unidad de filtro 349 se muestra "en bucle", las técnicas de la presente divulgación también se podrían usar con filtros posteriores, caso en el que los datos no filtrados (en lugar de los datos filtrados) se usarían con propósitos de predicción de datos en CU posteriores.

[0083] Para una serie de bloques de vídeo, tales como un fragmento o trama, la unidad de filtro 349 puede seleccionar conjuntos de filtros para cada entrada de una manera que promueva la calidad de vídeo. Por ejemplo, la unidad de filtro 349 puede seleccionar conjuntos de filtros de conjuntos de coeficientes predefinidos, o puede

definir filtros de forma adaptativa para promover la calidad de vídeo o una compresión mejorada. La unidad de filtro 349 puede seleccionar o definir uno o más conjuntos de filtros para una CU dada, de modo que se use(n) el/los mismo(s) conjunto(s) de filtros para píxeles de diferentes bloques de vídeo de esa CU. Para una trama, fragmento o LCU particular, la unidad de filtro 349 puede aplicar varios conjuntos de filtros a múltiples entradas, y la FSU 353 puede seleccionar el conjunto que produce el vídeo de mejor calidad o los niveles más altos de compresión. De forma alternativa, la FSU 353 puede entrenar un nuevo filtro analizando las autocorrelaciones y las correlaciones cruzadas entre múltiples entradas y una imagen original. Se puede determinar un nuevo conjunto de filtros, por ejemplo, resolviendo ecuaciones de Wiener-Hopt basadas en autocorrelaciones y correlaciones cruzadas. Independientemente de si se entrena un nuevo conjunto de filtros o si se selecciona un conjunto de filtros existente, la unidad de filtro 349 genera una sintaxis para su inclusión en el flujo de bits que posibilita que un descodificador también identifique el conjunto o conjuntos de filtros que se van a usar para la trama o fragmento particular.

[0084] De acuerdo con la presente divulgación, por cada píxel de una CU dentro de la serie de bloques de vídeo, la unidad de filtro 349 puede seleccionar qué filtro del conjunto de filtros se va a usar en base a dos o más métricas que cuantifican las propiedades asociadas con uno o más conjuntos de píxeles dentro de la CU. De esta manera, la FSU 353 puede determinar conjuntos de filtros para una unidad codificada de mayor nivel, tal como una trama o fragmento, mientras que la unidad de filtro 349 determina qué filtro(s) del/de los conjunto(s) se va(n) a usar para un píxel particular de una unidad codificada de menor nivel en base a las dos o más métricas asociadas a los píxeles de esa unidad codificada de menor nivel.

[0085] Se puede usar un conjunto de M filtros para cada entrada. Dependiendo de las preferencias de diseño, M puede ser, por ejemplo, tan pequeño como 2 o tan grande como 16, o incluso mayor. Una gran cantidad de filtros por entrada puede mejorar la calidad de vídeo, pero también puede incrementar la sobrecarga asociada con los conjuntos de señalización de filtros desde el codificador al descodificador. El conjunto de M filtros se puede determinar por la FSU 353 como se describe anteriormente y señalarse al descodificador para cada trama o fragmento. Se puede usar un mapa de segmentación para indicar cómo está segmentada una CU y si se va a filtrar o no una subunidad particular de la CU. El mapa de segmentación, por ejemplo, puede incluir para una CU una matriz de indicadores separados como se describe anteriormente, así como un bit adicional que señala si se va a filtrar cada sub-CU. Para cada entrada asociada a un píxel de una CU que se va a filtrar, se puede elegir un filtro específico del conjunto de filtros en base a dos o más métricas. Las combinaciones de valores para dos o más métricas se pueden indexar a los filtros particulares del conjunto de M filtros.

[0086] La FIG. 4A es un diagrama conceptual que ilustra rangos de valores para dos métricas indexadas a los filtros de un conjunto de filtros. El ejemplo particular de la FIG. 4A muestra ocho filtros (es decir, filtro 1, filtro 2... filtro 8), pero se pueden usar más o menos filtros de forma similar. La FIG. 4A muestra dos métricas que se podrían usar para seleccionar un filtro de acuerdo con las técnicas de la presente divulgación. Las dos métricas, por ejemplo, pueden cuantificar las propiedades de los datos de píxeles relacionados con la actividad no específica de la dirección (por ejemplo, un valor laplaciano modificado por suma) y la dirección, la actividad específica de la dirección y la detección de bordes, una métrica de dirección y una métrica de borde, una métrica de actividad horizontal y una métrica de actividad vertical u otras dos de dichas métricas. En algunos casos, se podrían usar tres o más métricas, caso en el que el diagrama conceptual de la FIG. 4A incluiría una tercera dimensión para asignar rangos de las métricas a los filtros del conjunto de filtros.

[0087] En el ejemplo de la FIG. 4A, una primera métrica (métrica 1) tiene cuatro rangos (rangos 1-1, 1-2, 1-3 y 1-4) y una segunda métrica (métrica 2) también tiene cuatro rangos (rangos 2-1, 2-2, 2-3 y 2-4). Por lo tanto, el ejemplo de la FIG. 4A tiene dieciséis combinaciones de rangos para la métrica 1 y la métrica 2. Como se puede ver en la FIG. 4A, sin embargo, cada combinación no está necesariamente asociada a un filtro único. La combinación de rango 1-1 y rango 2-1, así como las combinaciones 1-1 y 2-2, y 1-1 y 2-3, por ejemplo, están todas asignadas al filtro 1, en el ejemplo de la FIG. 4A. El filtro 4, por el contrario, solo se asigna a una combinación (1-1 y 2-4). Aunque los rangos de la FIG. 4A se muestran como relativamente iguales, los tamaños de los rangos pueden variar. Por ejemplo, en algunas implementaciones, el rango 1-1 puede englobar un rango de valores mayor que el rango 1-2. Además, aunque la FIG. 4A muestra que la métrica 1 y la métrica 2 tienen el mismo número de rangos, el número de rangos para una primera métrica y el número de rangos para una segunda métrica no necesariamente necesitan ser iguales. Si, por ejemplo, la métrica 1 es una métrica de varianza y la métrica 2 es una métrica de dirección, la métrica 1 podría usar ocho rangos mientras que la métrica 2 usa tres rangos.

[0088] En algunos ejemplos, los rangos de la métrica 1 y la métrica 2 pueden representar un espectro continuo de valores. Por ejemplo, si la métrica 1 es un valor laplaciano modificado por suma, el rango 1-2 puede corresponder a más actividad que el rango 1-1 pero menos actividad que el rango 1-3 y el rango 1-4 puede corresponder a más actividad que el rango 1-3. Dentro de un rango, la cantidad de actividad determinada para un píxel o grupo de píxeles particular se puede incrementar de forma similar a lo largo del eje de la métrica 1. En otros ejemplos, los rangos de la métrica 1 y la métrica 2 pueden no representar rangos reales, sino que pueden representar determinaciones discretas. Por ejemplo, si la métrica 2 es una métrica de dirección, el rango 1-1 puede corresponder a una determinación de dirección nula, el rango 2-2 puede corresponder a una determinación de dirección horizontal, el rango 2-3 puede corresponder a una determinación de dirección vertical y el rango 2-4 puede representar una determinación de dirección diagonal. Como se describirá con más detalle a continuación,

la dirección nula, dirección horizontal, dirección vertical y dirección diagonal pueden ser determinaciones discretas y, por tanto, los rangos para la métrica 2 podrían no representar un espectro continuo de valores de la misma manera que los rangos de la métrica 1 lo hacen.

[0089] La FIG. 4B es un diagrama conceptual que ilustra rangos de valores para una métrica de actividad y una métrica de dirección. En el ejemplo de la FIG. 4B, la métrica de dirección incluye tres determinaciones discretas (dirección nula, horizontal y vertical). Las técnicas para determinar la dirección nula, horizontal y vertical, así como las técnicas para determinar la actividad se explicarán con mayor detalle a continuación. El ejemplo particular de la FIG. 4B muestra seis filtros (es decir, filtro 1, filtro 2... filtro 6), pero se pueden usar más o menos filtros de forma similar. Como se puede ver en la FIG. 4B, las dos métricas (actividad y dirección) crean 15 combinaciones, identificadas como combinaciones de 421 a 435. En algunos casos, sin embargo, también se pueden usar combinaciones adicionales que no se muestran explícitamente en la FIG. 4B. Por ejemplo, una combinación correspondiente a actividad nula puede ser una 16.^a combinación que también tiene un filtro correspondiente.

[0090] La unidad de filtro 349 puede almacenar una asignación de filtros a combinaciones de rangos de dos o más métricas, tales como las asignaciones de ejemplo de las FIGS. 4A y 4B, y usar la asignación para determinar qué filtro de un conjunto de filtros aplicar a un píxel o grupo de píxeles particular en una CU. La asignación de filtros a combinaciones de rangos de dos o más métricas, por ejemplo, se puede determinar por la unidad de filtro 349 como parte del proceso de selección de filtro descrito anteriormente. Independientemente de cómo se determine la asignación, la unidad de filtro 349 puede generar información que permita que un descodificador reconstruya la asignación. Esta información se puede incluir en el flujo de bits codificado para señalar la asignación de combinaciones de rangos a filtros. La asignación de combinaciones a rangos señalados puede asignar combinaciones de rangos para filtrar identificaciones ID. Los coeficientes reales para un filtro particular se podrían señalar por separado.

[0091] Para generar esta información, la unidad de filtro 349 determina en primer lugar un orden de transmisión para las combinaciones. El orden de transmisión en general se refiere al orden en que se señalarán los filtros para combinaciones de rangos. Usando la FIG. 4A como ejemplo, la unidad de filtro 349 podría usar un orden de transmisión de izquierda a derecha, de arriba a abajo, donde en primer lugar se señala el filtro para la combinación 401, en segundo lugar se señala el filtro para la combinación 402 y las combinaciones restantes se señalan en el orden de 403 => 404 => 405 => 406 => 407 => 408 => 409 => 410 => 411 => 412 => 413 => 414 => 415 => 416. La unidad de filtro 349 también podría usar un orden de transmisión de arriba a abajo en zigzag, donde los filtros para las combinaciones se señalan en el orden de 401 => 402 => 403 => 404 => 408 => 407 => 406 => 405 => 409 => 410 => 411 => 412 => 416 => 415 => 414 => 413. La unidad de filtro 349 también podría usar un orden de transmisión de arriba a abajo, de izquierda a derecha, donde los filtros para combinaciones se señalan en el orden de 401 => 405 => 409 => 413 => 402 => 406 => 410 => 414 => 403 => 407 => 411 => 415 => 404 => 408 => 412 => 416. La unidad de filtro 349 también podría usar un orden de transmisión de izquierda a derecha en zigzag donde los filtros para combinaciones se señalan en el orden de 401 => 405 => 409 => 413 => 414 => 410 => 406 => 402 => 403 => 407 => 411 => 415 => 416 => 412 => 408 => 404. Con referencia a la FIG. 4B, la unidad de filtro 349 puede usar un orden de transmisión de izquierda a derecha, de abajo a arriba, de modo que el orden de transmisión sea 421 => 422 => 423 => 424 => 425 => 426 => 427 => 428 => 429 => 430 => 431 => 432 => 433 => 434 => 435. Como se puede imaginar, estos son solo unos pocos de los muchos órdenes de transmisión que son posibles.

[0092] De acuerdo con una técnica de la presente divulgación, la unidad de filtro 349 puede usar una serie de contraseñas para señalar la asignación a un descodificador. Por ejemplo, la unidad de filtro 349 puede generar una primera contraseña para indicar si una combinación actual que se está descodificando se asigna al mismo filtro que la combinación descodificada lo más recientemente que comparte el mismo rango para la primera métrica. Si una combinación actual que se está descodificando se asigna al mismo filtro que la combinación descodificada lo más recientemente que comparte el mismo rango para la segunda métrica, entonces la unidad de filtro 349 puede generar una segunda contraseña en lugar de la primera contraseña. Si una combinación actual que se está descodificando no se asigna al mismo filtro que cualquiera de estas combinaciones descodificadas lo más recientemente, entonces la unidad de filtro 349 puede generar una tercera contraseña, en lugar de la primera contraseña o la segunda contraseña, que indica el filtro correspondiente a la combinación actual se está descodificando. La primera y la segunda contraseña del ejemplo actual pueden ser relativamente cortas en comparación con la tercera contraseña. Por ejemplo, la primera contraseña y la segunda contraseña podrían tener cada una dos bits (por ejemplo, 00 y 01, respectivamente), mientras que la tercera contraseña tiene más bits (un primer bit de 1, más bits adicionales). En este contexto particular, una combinación actual que se está descodificando o una combinación previa que se está descodificando se refiere a la parte de los procesos de codificación y descodificación donde la asignación de filtros a combinaciones de rango está señalada por un codificador o construida por un descodificador, y no necesariamente a una transmisión o descodificación de la combinación por sí misma.

[0093] Ahora se darán ejemplos de las técnicas descritas anteriormente con referencia a la FIG. 4A y a un orden de transmisión de arriba a abajo, de izquierda a derecha. Si, por ejemplo, la combinación 407 es la combinación que se está descodificando actualmente, entonces la combinación 406 es la combinación descodificada lo más

recientemente que comparte el mismo rango para la métrica 1, y la combinación 403 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 2. Si la combinación 407 se asigna al mismo filtro (filtro 7 en la FIG. 4A) que la combinación descodificada lo más recientemente que comparte el mismo rango para una segunda métrica (es decir, el rango 2-3 para la métrica 2), entonces la unidad de filtro 349 puede transmitir una segunda contraseña (por ejemplo, 01) para indicar que la combinación actual que se está descodificando (combinación 407) se asigna al mismo filtro que la combinación descodificada lo más recientemente que comparte el mismo rango para una segunda métrica (combinación 403).

[0094] Si, por ejemplo, la combinación 410 es la combinación actual que se está descodificando, entonces la combinación 409 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 1, y la combinación 406 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 2. Si la combinación 410 se asigna al mismo filtro (filtro 2 en la FIG. 4A) que la combinación descodificada lo más recientemente que comparte el mismo rango para una primera métrica (es decir, el rango 1-2 para la métrica 1), entonces la unidad de filtro 349 puede transmitir una primera contraseña (por ejemplo, 00) para indicar que la combinación actual que se está descodificando (combinación 410) se asigna al mismo filtro (filtro 2) que la combinación descodificada lo más recientemente que comparte el mismo rango para una primera métrica (combinación 409).

[0095] Si, por ejemplo, la combinación 411 es la combinación actual que se está descodificando, entonces la combinación 410 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 1, y la combinación 407 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 2. Si la combinación 411 no se asigna al mismo filtro que la combinación 410 o bien la combinación 407, entonces la unidad de filtro 349 puede transmitir una tercera contraseña (por ejemplo, 1 + bits adicionales) para indicar que la combinación actual que se está descodificando (combinación 411) se asigna a un filtro diferente (filtro 3) que tanto la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 1 como la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 2.

[0096] Para aquellas combinaciones actuales donde una combinación que comparte el mismo rango para la métrica 1 o una combinación que comparte el mismo rango para la métrica 2 aún no se han descodificado, entonces esas opciones se pueden considerar no disponibles o bien se pueden reemplazar por una combinación diferente. Si, por ejemplo, la combinación 409 es la combinación actual que se va a descodificar, entonces la combinación 405 es la combinación descodificada lo más recientemente que comparte el mismo rango para la métrica 2, pero aún no se ha descodificado ninguna combinación que comparta un rango para la métrica 1. En dichos casos, se puede suponer que la combinación descodificada lo más recientemente que comparte un rango para la métrica 1 no se asigna al mismo filtro que la combinación actual que se está descodificando. Por tanto, en este caso, la primera contraseña no se usará para la combinación 409. De forma alternativa, la combinación que comparte un rango para la métrica 1 se puede reemplazar por otra combinación, tal como la combinación descodificada lo más recientemente o una combinación previamente descodificada diferente. En un caso de este tipo, la combinación descodificada lo más recientemente antes de la combinación 409 sería la combinación 408. Por tanto, si la combinación 408 se asigna al mismo filtro que la combinación 409, entonces la unidad de filtro 349 puede generar la primera contraseña. Se pueden usar técnicas análogas para aquellas combinaciones donde aún no se ha descodificado una combinación previa que comparte un rango común para la métrica 1.

[0097] Para la primera combinación en un orden de transmisión (por ejemplo, la combinación 401 en el ejemplo de la FIG. 4A), donde ni una combinación que comparte el mismo rango para la métrica 1 ni una combinación que comparte el mismo rango para la métrica 2 se han descodificado, la unidad de filtro 349 puede generar una contraseña que indica el filtro que se asigna a la primera combinación. El filtro, por ejemplo, se puede señalar usando la tercera contraseña o se puede señalar usando una técnica diferente, caso en el que las técnicas descritas en la presente divulgación podrían comenzar con la segunda combinación en un orden de transmisión o una combinación posterior.

[0098] De acuerdo con otra técnica de la presente divulgación, la unidad de filtro 349 puede usar una serie de contraseñas para señalar la asignación a un descodificador. En algunas implementaciones, la unidad de filtro 349 puede generar una primera contraseña para indicar si una combinación actual que se está descodificando se asigna al mismo filtro que la combinación descodificada lo más recientemente que comparte el mismo rango para la primera métrica. Si una combinación actual que se está descodificando no se asigna al mismo filtro que la combinación descodificada lo más recientemente que comparte ese rango para la primera métrica, entonces la unidad de filtro 349 puede generar una segunda contraseña, en lugar de la primera contraseña, que indica el filtro que se asigna a la combinación actual que se está descodificando. En este ejemplo, la primera contraseña puede ser relativamente corta en comparación con la segunda contraseña. Por ejemplo, la primera contraseña podría tener un bit (por ejemplo, 0), mientras que la segunda contraseña tiene más bits (por ejemplo, un primer bit de 1, más bits adicionales). A diferencia de la técnica previa donde se podría generar una contraseña corta si una combinación actual se asigna al mismo filtro que una combinación previamente descodificada que comparte el mismo rango para la métrica 1 o bien la métrica 2, esta técnica incluye solo generar una contraseña corta si la combinación actual se asigna al mismo filtro que una combinación previamente descodificada que comparte el

mismo rango para la métrica 1. Por tanto, incluso si la combinación actual se asigna al mismo filtro que una combinación previamente descodificada que comparte el mismo rango para la métrica 2, la unidad de filtro 349 todavía genera una segunda contraseña (por ejemplo, 1 + bits adicionales). Aunque la presente divulgación usa la métrica 1 con propósitos de explicación, también se pueden aplicar las mismas técnicas usando solo la métrica 2.

[0099] De acuerdo con aún otra técnica de la presente divulgación, la unidad de filtro 349 puede usar una serie diferente de contraseñas para señalar la asignación a un descodificador. Por ejemplo, la unidad de filtro 349 puede generar una primera contraseña para indicar si una combinación actual que se está descodificando se asigna al mismo filtro que la combinación descodificada lo más recientemente, independientemente de qué rango, si hubiere alguno, tiene en común la combinación actual con la combinación previamente descodificada. Si la combinación actual que se está descodificando no se asigna al mismo filtro que la combinación descodificada lo más recientemente, entonces la unidad de filtro 349 puede generar una segunda contraseña que identifica el filtro que se asigna a la combinación actual. En esta implementación particular, la primera contraseña puede ser relativamente corta en comparación con la segunda contraseña. Por ejemplo, la primera contraseña podría tener un bit (por ejemplo, 0), mientras que la segunda contraseña tiene más bits (por ejemplo, un primer bit de 1, más bits adicionales).

[0100] De nuevo, usando el ejemplo de la FIG. 4A y un orden de transmisión de arriba a abajo, de izquierda a derecha, la combinación 401 sería la combinación descodificada lo más recientemente si la combinación 402 se está descodificando actualmente, la combinación 402 sería la combinación descodificada lo más recientemente si la combinación 403 es la combinación actual, y así sucesivamente. 404 sería la combinación descodificada lo más recientemente si la combinación 405 es la combinación actual que se está descodificando. Por tanto, la unidad de filtro 349 puede generar la primera contraseña si la combinación 402 se asigna al mismo filtro que la combinación 401, si la combinación 403 se asigna al mismo filtro que la combinación 402, etc. De otro modo, la unidad de filtro 349 puede generar la segunda contraseña que identifica el filtro que se asigna a la combinación actual.

[0101] De acuerdo con aún otra técnica de la presente divulgación, la unidad de filtro 349 puede usar dos contraseñas para señalar la asignación de los filtros a las combinaciones. Se puede usar una primera contraseña, tal como un "0", para señalar que una combinación actual usa el mismo filtro que una combinación previa. Se puede usar una segunda contraseña, tal como un "1", para señalar que una combinación actual tiene un filtro diferente que la combinación previa. La segunda contraseña, sin embargo, no necesita identificar un nuevo filtro. En cambio, el nuevo filtro se puede determinar en base al orden de transmisión para las clases y el orden en que se transmiten los coeficientes de filtro. Usando el orden de transmisión de izquierda a derecha, de abajo a arriba, descrito anteriormente para la FIG. 4B como ejemplo, se podrían transmitir las contraseñas en consecuencia: 421 (0) => 422 (0) => 423 (1) => 424 (0) => 425 (0) => 426 (0) => 427 (0) => 428 (1) => 429 (0) => 430 (0) => 431 (0) => 432 (1) => 433 (0) => 434 (0) => 435 (0), representando el número entre paréntesis la contraseña para esa combinación. En este ejemplo, se asignarían las combinaciones 421-422 a un primer filtro, las combinaciones 423-427 a un segundo filtro, las combinaciones 428-431 a un tercer filtro y las combinaciones 432-435 a un cuarto filtro. Los coeficientes para el primer filtro, el segundo filtro, el tercer filtro y el cuarto filtro pueden corresponder al orden en que se señalan los conjuntos de coeficientes de filtro, donde el primer conjunto de coeficientes de filtro señalado corresponde al primer filtro, el segundo conjunto de coeficientes de filtro señalado corresponde al segundo filtro, y así sucesivamente. La determinación de un orden para transmitir conjuntos de coeficientes de filtro se analiza con más detalle a continuación.

[0102] Las diversas técnicas descritas en la presente divulgación para señalar una asignación de filtros a combinaciones de rangos no son alternativas mutuamente excluyentes, sino que se pueden usar en conjunto entre sí. Por ejemplo, en algunas implementaciones, se podrían señalar determinadas combinaciones usando una primera técnica, mientras que otras combinaciones se señalan usando una segunda técnica. Como un ejemplo, cuando una combinación que comparte el mismo rango para la métrica 1 o una combinación que comparte el mismo rango para la métrica 2 aún no se han descodificado (por ejemplo, las combinaciones 402, 403, 404, 405, 409 y 413), entonces la unidad de filtro 349 puede usar una primera técnica. Cuando se han descodificado tanto una combinación que comparte el mismo rango para la métrica 1 como una combinación que comparte el mismo rango para la métrica 2 (por ejemplo, las combinaciones 406, 407, 408, 410, 411, 412, 414, 415 y 416), entonces se podría usar una segunda técnica. Adicionalmente, las contraseñas usadas para cualquiera de la primera, segunda y tercera contraseñas descritas anteriormente pueden ser cualquiera de contraseñas de longitud fija, contraseñas de longitud variable o contraseñas de longitud variable adaptativas al contexto.

[0103] Además de generar información que permite a un descodificador reconstruir la asignación de filtros a combinaciones de rangos, la unidad de filtro 349 también genera información que permite a un descodificador reconstruir los filtros por sí mismos. La reconstrucción de los filtros incluye la reconstrucción de los coeficientes de filtro de los filtros. Como se describirá con más detalle a continuación, la unidad de filtro 349 puede usar técnicas de codificación diferencial para señalar los coeficientes de filtro. Para usar la técnica de codificación diferencial, la unidad de filtro 349 determina un orden en el que se señalan los conjuntos de coeficientes de filtro.

[0104] Como parte de la determinación del orden, la unidad de filtro 349 determina una identificación de combinación (ID) que representa un valor secuencial para cada combinación de rangos. Usando la FIG. 4A como

ejemplo, a las combinaciones se les podrían asignar las ID de combinación que representan valores secuenciales en un orden de izquierda a derecha, de arriba a abajo, caso en el que a la combinación 401 se le asignaría el primer valor secuencial, a la combinación 402 se le asignaría el segundo valor secuencial y a las combinaciones restantes se les asignarían valores secuenciales en el orden de 403 => 404 => 405 => 406 => 407 => 408 => 409 => 410 => 411 => 412 => 413 => 414 => 415 => 416. La unidad de filtro 349 también podría asignar las ID de combinación usando un orden de arriba a abajo en zigzag donde a las combinaciones se les asignarían las ID de combinación con valores secuenciales que están en el orden de 401 => 402 => 403 => 404 => 408 => 407 => 406 => 405 => 409 => 410 => 411 => 412 => 416 => 415 => 414 => 413. La unidad de filtro 349 también podría asignar las ID de combinación usando un orden de arriba a abajo, de izquierda a derecha, donde a las combinaciones se les asignan las ID de combinación con valores secuenciales que están en el orden de 401 => 405 => 409 => 413 => 402 => 406 => 410 => 414 => 403 => 407 => 411 => 415 => 404 => 408 => 412 => 416. La unidad de filtro 349 también podría usar un orden de izquierda a derecha en zigzag donde a las combinaciones se les asignan las ID de combinación con valores secuenciales en un orden de 401 => 405 => 409 => 413 => 414 => 410 => 406 => 402 => 403 => 407 => 411 => 415 => 416 => 412 => 408 => 404. Como se puede imaginar, estos son solo unos pocos de los muchos órdenes que se podrían usar. Además, cualquiera de los órdenes descritos podría ser del menor al mayor o bien del mayor al menor.

[0105] Después de que la unidad de filtro 349 haya determinado la asignación de filtros a combinaciones de rango, la unidad de filtro 349 puede identificar agrupaciones de combinaciones de rango que se asignan al mismo filtro. Usando la FIG. 4A como ejemplo, las agrupaciones serían como sigue.

Grupo de filtro 1: combinaciones 413, 414 y 415

Grupo de filtro 2: combinaciones 409, 410

Grupo de filtro 3: combinaciones 411 y 412

Grupo de filtro 4: combinación 416

Grupo de filtro 5: combinaciones 401 y 405

Grupo de filtro 6: combinaciones 402 y 406

Grupo de filtro 7: combinaciones 403 y 407

Grupo de filtro 8: combinaciones 404 y 408.

[0106] A continuación, la unidad de filtro 349 puede asignar a cada grupo una ID de grupo, y la ID de grupo puede representar un valor secuencial. Las ID de grupo se pueden asignar a los grupos en base a los valores secuenciales asociados a las combinaciones que comprenden el grupo. Por ejemplo, al grupo que tiene la combinación con el valor secuencial asociado inferior en base a las ID de combinación, se le podría asignar la ID de grupo con el valor secuencial inferior. De los grupos restantes, al grupo restante que tiene la combinación con el valor secuencial asociado inferior se le puede asignar la ID de grupo con el siguiente valor secuencial inferior. Este proceso se puede repetir hasta que a todos los grupos se les haya asignado una ID de grupo. En algunas implementaciones, las ID de grupo se pueden asignar en base a las combinaciones con los valores secuenciales asociados superiores en lugar de los inferiores. En algunas implementaciones, al grupo que tiene la combinación con el valor secuencial asociado inferior en base a las ID de combinación, se les podría asignar la ID de grupo con el valor secuencial superior, o viceversa.

[0107] De nuevo, usando la FIG. 4A como ejemplo, y suponiendo que a las combinaciones 401-416 se les asignan las ID de combinación con valores secuenciales en un orden de izquierda a derecha, de arriba a abajo, entonces la unidad de filtro 349 puede asignar las ID de grupo a los grupos de filtros, como se muestra a continuación en la tabla 1.

Tabla 1.

Nombre de grupo	Combinaciones en el grupo	Combinación con el valor secuencial inferior	ID de grupo
Grupo de filtro 1	413, 414, 415	413	7
Grupo de filtro 2	409, 410	409	5
Grupo de filtro 3	411, 412	411	6
Grupo de filtro 4	416	416	8
Grupo de filtro 5	401, 405	401	1
Grupo de filtro 6	402, 406	402	2

Grupo de filtro 7	403, 407	403	3
Grupo de filtro 8	404, 408	404	4

[0108] En el ejemplo de la FIG. 4A, mostrado en la tabla 1, la unidad de filtro 349 asigna al grupo de filtro 5 la ID del grupo con el valor secuencial inferior porque el grupo de filtro 5 incluye la combinación de rango con el valor secuencial inferior (es decir, la combinación 401). La unidad de filtro 349 asigna al grupo de filtro 6 la ID de grupo con el segundo valor secuencial inferior porque, de los grupos de filtro restantes (es decir, todos los grupos excluyendo el grupo de filtro 5), el grupo de filtro 6 incluye la combinación de rango con el segundo valor secuencial inferior (es decir, la combinación 402). La unidad de filtro 349 asigna al grupo de filtro 7 la ID de grupo con el tercer valor secuencial inferior porque, de los grupos de filtro restantes (es decir, todos los grupos de filtro excluyendo el grupo de filtro 5 y el grupo de filtro 6), el grupo de filtro 7 incluye la combinación de rango con el valor secuencial inferior (es decir, la combinación 403). La unidad de filtro 349 asigna al grupo de filtro 8 la ID de grupo con el cuarto valor secuencial inferior porque, de los grupos de filtro restantes (es decir, todos los grupos de filtro excluyendo el grupo de filtro 5, el grupo de filtro 6 y el grupo de filtro 7), el grupo de filtro 8 incluye la combinación de rango con el cuarto valor secuencial inferior (combinación 404). La unidad de filtro 349 asigna al grupo de filtro 2 la ID de grupo con el quinto valor secuencial inferior porque, de los grupos de filtro restantes (es decir, excluyendo el grupo de filtro 5, el grupo de filtro 6, el grupo de filtro 7 y el grupo de filtro 8), el grupo de filtro 2 incluye la combinación de rango con el valor secuencial inferior (combinación 409). La unidad de filtro 349 asigna al grupo de filtro 3 la ID de grupo con el sexto valor secuencial inferior porque, de los grupos de filtro restantes (es decir, excluyendo el grupo de filtro 5, el grupo de filtro 6, el grupo de filtro 7, el grupo de filtro 8 y el grupo de filtro 2), el grupo de filtro 3 incluye la combinación de rango con el valor secuencial inferior (combinación 411). La unidad de filtro 349 asigna al grupo de filtro 1 la ID de grupo con el séptimo valor secuencial inferior porque, de los grupos de filtro restantes (es decir, excluyendo el grupo de filtro 5, el grupo de filtro 6, el grupo de filtro 7, el grupo de filtro 8, el grupo de filtro 2 y el grupo de filtro 3), el grupo de filtro 1 incluye la combinación de rango con el valor secuencial inferior (combinación 413). Finalmente, la unidad de filtro 349 asigna al grupo de filtro 4, el grupo de filtro restante final, la ID de grupo con el valor secuencial superior (8 en este ejemplo particular).

[0109] En base a las ID de grupo de filtros, la unidad de filtro 349 determina un orden en el que se señalan los coeficientes de filtro de un filtro. De nuevo, usando el ejemplo de la FIG. 4A y la tabla 1, en primer lugar la unidad de filtro 349 señala el coeficiente para el filtro 5, a continuación el coeficiente para el filtro 6, a continuación el coeficiente para el filtro 7, a continuación el coeficiente para el filtro 8, a continuación el coeficiente para el filtro 2, a continuación el coeficiente para el filtro 3, a continuación el coeficiente para el filtro 1 y finalmente el coeficiente para el filtro 4. Usando técnicas de codificación diferencial, como se describe en la presente divulgación, la unidad de filtro 349 puede codificar los coeficientes para el filtro 6 como información de diferencia en relación con los coeficientes de filtro del filtro 5, codificar los coeficientes para el filtro 7 como información de diferencia en relación con los coeficientes de filtro para el filtro 6 y así sucesivamente, en base al orden secuencial de las ID de grupo.

[0110] La asignación de dos o más métricas para las entradas a los filtros se puede implementar de múltiples maneras. Por ejemplo, en algunas implementaciones cada entrada podría tener un conjunto de filtros único, mientras que en algunas implementaciones las entradas comparten un conjunto de filtros común. Adicionalmente, en algunas implementaciones, se podrían usar dos o más métricas para cada entrada para identificar un filtro particular para cada entrada. Sin embargo, en otras implementaciones, se podrían usar dos o más métricas para una única entrada para identificar filtros para todas las entradas. Aún en otras implementaciones, se podrían usar dos o más métricas para una primera entrada para identificar un filtro para una segunda entrada diferente.

[0111] De acuerdo con la presente divulgación, la unidad de filtro 349 puede realizar técnicas de codificación con respecto a la información del filtro que puede reducir la cantidad de datos necesarios para codificar y transmitir la información del filtro desde el codificador 350 a otro dispositivo. De nuevo, para cada trama o fragmento, la unidad de filtro 349 puede definir o seleccionar uno o más conjuntos de coeficientes de filtro que se van a aplicar a los píxeles de las CU para esa trama o fragmento. La unidad de filtro 349 aplica los coeficientes de filtro para filtrar bloques de vídeo de tramas de vídeo reconstruidas almacenadas en la memoria 334, que se pueden usar para codificación predictiva consecuente con el filtrado en bucle. La unidad de filtro 349 puede codificar los coeficientes de filtro como información del filtro, que se reenvía a la unidad de codificación por entropía 346 para su inclusión en el flujo de bits codificado.

[0112] Adicionalmente, las técnicas de la presente divulgación pueden aprovechar el hecho de que algunos de los coeficientes de filtro definidos o seleccionados por la FSU 353 pueden ser muy similares a otros coeficientes de filtro aplicados con respecto a los píxeles de las CU de otra trama o fragmento. Se puede aplicar el mismo tipo de filtro para diferentes tramas o fragmentos (por ejemplo, el mismo soporte de filtro), pero los filtros pueden ser diferentes en términos de valores de coeficientes de filtro asociados a los diferentes índices del soporte de filtro. En consecuencia, para reducir la cantidad de datos necesarios para transmitir dichos coeficientes de filtro, la unidad de filtro 349 puede codificar de forma predictiva uno o más coeficientes de filtro que se van a usar para el filtrado en base a los coeficientes de filtro de otra CU, aprovechando potencialmente las similitudes entre los coeficientes de filtro. Sin embargo, en algunos casos, puede ser más deseable codificar los coeficientes de filtro directamente, por ejemplo, sin usar ninguna predicción. Se pueden usar diversas técnicas, tales como las técnicas que

aprovechan el uso de una métrica de actividad para definir cuándo codificar los coeficientes de filtro usando técnicas de codificación predictiva y cuándo codificar los coeficientes de filtro directamente sin ninguna codificación predictiva, para comunicar eficazmente los coeficientes de filtro a un decodificador. Adicionalmente, también se puede imponer la simetría de modo que se pueda usar un subconjunto de coeficientes (por ejemplo, 5, -2, 10) conocidos por el decodificador para definir el conjunto completo de coeficientes (por ejemplo, 5, -2, 10, 10, -2, 5). Se puede imponer la simetría en los contextos de codificación tanto directa como predictiva.

[0113] Como se describe anteriormente, el codificador de vídeo 350 representa un ejemplo de un codificador de vídeo configurado para determinar una primera métrica para un grupo de píxeles dentro de un bloque de píxeles, determinar una segunda métrica para el grupo de píxeles, determinar un filtro en base a la primera métrica y la segunda métrica y generar una imagen filtrada aplicando el filtro al grupo de píxeles. El codificador de vídeo 350 también representa un ejemplo de un codificador de vídeo configurado para determinar una primera métrica para un bloque de píxeles, en el que la primera métrica se determina en base a una comparación de un subconjunto de los píxeles en el bloque con otros píxeles en el bloque; determinar una segunda métrica para el bloque de píxeles; determinar un filtro en base a la primera métrica y la segunda métrica; y, generar una imagen filtrada aplicando el filtro al bloque de píxeles.

[0114] Como se describe anteriormente, el codificador de vídeo 350 también representa un ejemplo de un codificador de vídeo configurado para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica, en el que cada combinación de rango tiene una identificación (ID) de combinación de rango exclusiva, en el que cada ID de combinación de rango exclusiva corresponde a un valor secuencial para una combinación de rango; asignar las ID de grupo exclusivas a grupos de combinaciones de rango en base a los valores secuenciales para las combinaciones de rango, en el que cada ID de grupo exclusiva corresponde a un valor secuencial para un grupo; y, codificar conjuntos de coeficientes de filtro correspondientes para los filtros en base a las ID de grupo exclusivas. El codificador de vídeo 350 puede codificar los conjuntos de coeficientes de filtro señalando los conjuntos de coeficientes de filtro en un flujo de bits codificado en un orden que se selecciona en base a los valores secuenciales de las ID de grupo exclusivas. El codificador de vídeo 350 puede señalar los conjuntos de coeficientes de filtro usando técnicas de codificación diferencial.

[0115] Como se describió anteriormente, el codificador de vídeo 350 también representa un ejemplo de un codificador de vídeo configurado para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango de valores para una primera métrica y un rango de valores para una segunda métrica; generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que comprende el mismo rango de valores para la primera métrica; generar una segunda contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que comprende el mismo rango de valores para la segunda métrica; y, generar una tercera contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa que comprende el mismo rango de valores para la primera métrica y la combinación de rango previa que comprende el mismo rango de valores para la segunda métrica. El codificador de vídeo 350 también representa un ejemplo de un codificador de vídeo configurado para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa; y, generar una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente que la combinación de rango previa, en el que la segunda contraseña identifica un filtro asignado a la combinación de rango actual.

[0116] La FIG. 5 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo 560, que decodifica una secuencia de vídeo que se codifica de la manera descrita en el presente documento. La secuencia de vídeo recibida puede comprender un conjunto codificado de tramas de imágenes, un conjunto de fragmentos de trama, un grupo de imágenes (GOP) codificado comúnmente o una amplia variedad de tipos de series de bloques de vídeo que incluyen bloques de vídeo codificado y sintaxis para definir cómo decodificar dichos bloques de vídeo.

[0117] El decodificador de vídeo 560 incluye una unidad de decodificación por entropía 552, que realiza la función de decodificación recíproca de la codificación realizada por la unidad de codificación por entropía 346 de la FIG. 3. En particular, la unidad de decodificación por entropía 552 puede realizar la decodificación CAVLC o CABAC, o cualquier otro tipo de decodificación por entropía usada por el codificador de vídeo 350. Los bloques de vídeo decodificados por entropía en un formato seriado unidimensional se pueden explorar de forma inversa para convertir uno o más vectores unidimensionales de coeficientes de nuevo a un formato de bloque bidimensional. El número y tamaño de los vectores, así como el orden de exploración definido para los bloques de vídeo pueden definir cómo se reconstruye el bloque bidimensional. La sintaxis de predicción decodificada por entropía se puede enviar desde la unidad de decodificación por entropía 552 al módulo de predicción 554, y la información del filtro decodificada por entropía se puede enviar desde la unidad de decodificación por entropía 552 a la unidad de filtro 559.

[0118] El descodificador de vídeo 560 también incluye un módulo de predicción 554, una unidad de cuantificación inversa 556, una unidad de transformada inversa 558, una memoria y un sumador 564. Además, el descodificador de vídeo 560 también incluye un filtro de eliminación de efecto pixelado 557 que filtra la salida del sumador 564. Consecuente con la presente divulgación, la unidad de filtro 559 puede recibir información del filtro descodificada por entropía que incluye uno o más filtros que se van a aplicar a una o más entradas. Aunque no se muestra en la FIG. 5, el filtro de eliminación de efecto pixelado 557 también puede recibir información del filtro descodificada por entropía que incluye uno o más filtros que se van a aplicar.

[0119] Los filtros aplicados por la unidad de filtro 559 se pueden definir por conjuntos de coeficientes de filtro. La unidad de filtro 559 se puede configurar para generar los conjuntos de coeficientes de filtro en base a la información del filtro recibida desde la unidad de descodificación por entropía 552. La información del filtro puede incluir una sintaxis de descripción de filtro que identifica un número máximo de filtros en un conjunto de filtros y/o una forma de filtros en un conjunto de filtros, por ejemplo. La sintaxis de descripción de filtro se puede incluir en un encabezado de una serie de bloques de vídeo, por ejemplo, un encabezado LCU, un encabezado de trama, un encabezado de fragmento, un encabezado GOP, un encabezado de secuencia o similares. En otros ejemplos, la sintaxis de descripción de filtro se podría incluir en un pie de página u otra estructura de datos. En base a la sintaxis de descripción de filtro, la unidad de filtro 559 puede reconstruir el conjunto de filtros usados en el codificador.

[0120] La información del filtro también puede incluir una sintaxis de señalización adicional que le señala al descodificador la manera de codificación usada para cualquier conjunto dado de coeficientes. En algunas implementaciones, la información del filtro, por ejemplo, puede incluir rangos para dos o más métricas para las que se debe usar cualquier conjunto de coeficientes dado. Tras la descodificación de los filtros, la unidad de filtro 559 puede filtrar los valores de píxel de los bloques de vídeo descodificados en base a uno o más conjuntos de coeficientes de filtro y la sintaxis de señalización que incluye los rangos para los que se deben usar los diferentes conjuntos de coeficientes de filtro.

[0121] La unidad de filtro 559 puede recibir en el flujo de bits uno o más elementos de sintaxis que indican un conjunto de filtros para cada trama o fragmento, así como una asignación de filtros a las dos o más métricas. Por ejemplo, si un codificador usa la asignación de rangos para métricas a los filtros mostrados en la FIG. 4A, entonces el codificador señalará esta asignación o bien transmitirá los datos para permitir que la unidad de filtro 559 reconstruya esta asignación. Independientemente de si esta asignación está o no explícitamente señalada, la unidad de filtro 559 puede mantener la misma asignación de filtros a combinaciones de rangos como se usa por el codificador.

[0122] Como se menciona anteriormente, la unidad de filtro 559 genera una asignación en base a la información del filtro señalada en el flujo de bits. En base a esta asignación, la unidad de filtro 559 puede determinar grupos y asignar las ID de grupo a los grupos de la misma manera descrita anteriormente en relación con la unidad de filtro 349. Usando estas ID de grupo, la unidad de filtro 559 puede asociar los coeficientes de filtro recibidos con

[0123] Para cada CU dentro de la trama o fragmento, la unidad de filtro 559 puede calcular una o más métricas asociadas con los píxeles descodificados de una CU para múltiples entradas (es decir, PI, EI, pRI y RI) para determinar qué filtro(s) del/de los conjunto(s) aplicar a cada entrada. De forma alternativa, la unidad de filtro 559 puede calcular una o más métricas para una única entrada, tal como pRI o RI. La unidad de filtro 559 determina qué filtro aplicar en base a las métricas determinadas para un píxel o grupo de píxeles particular. Usando un valor laplaciano modificado por suma y una dirección como ejemplos para la métrica 1 y la métrica 2 y usando las asignaciones mostradas en la FIG. 4A como ejemplo, si la unidad de filtro 559 determina que un píxel o grupo de píxeles tiene un valor laplaciano modificado por suma en el rango 1-2 y una dirección correspondiente al rango 2-3, entonces la unidad de filtro 559 puede aplicar el filtro 2 a ese píxel o grupo de píxeles. Si la unidad de filtro 559 determina que un píxel o grupo de píxeles tiene un valor laplaciano modificado por suma en el rango 1-4 y una dirección correspondiente al rango 2-2, entonces la unidad de filtro 559 puede aplicar el filtro 6 a ese píxel o grupo de píxeles y así sucesivamente. El filtro en general puede asumir cualquier tipo de forma o disposición de soporte de filtro. El soporte de filtro se refiere a la forma del filtro con respecto a un píxel dado que se está filtrando, y los coeficientes de filtro pueden definir la ponderación aplicada a los valores de píxel contiguos de acuerdo con el soporte de filtro. De acuerdo con las técnicas de la presente divulgación, los datos de sintaxis se pueden incluir en el flujo de bits para señalar al descodificador cómo se codificaron los filtros (por ejemplo, cómo se codificaron los coeficientes de filtro), así como los rangos de la métrica de actividad para la que se deben usar diferentes filtros.

[0124] Para cada CU dentro de la trama o fragmento, la unidad de filtro 559 puede calcular una o más métricas asociadas con los píxeles descodificados de una CU para múltiples entradas (es decir, PI, EI, pRI y RI) para determinar qué filtro(s) del/de los conjunto(s) aplicar a cada entrada. De forma alternativa, la unidad de filtro 559 puede calcular una o más métricas para una única entrada, tal como pRI o RI. La unidad de filtro 559 determina qué filtro aplicar en base a las métricas determinadas para un píxel o grupo de píxeles particular. Usando un valor laplaciano modificado por suma y una dirección como ejemplos para la métrica 1 y la métrica 2 y usando las asignaciones mostradas en la FIG. 4A como ejemplo, si la unidad de filtro 559 determina que un píxel o grupo de píxeles tiene un valor laplaciano modificado por suma en el rango 1-2 y una dirección correspondiente al rango 2-3, entonces la unidad de filtro 559 puede aplicar el filtro 2 a ese píxel o grupo de píxeles. Si la unidad de filtro 559

determina que un píxel o grupo de píxeles tiene un valor laplaciano modificado por suma en el rango 1-4 y una dirección correspondiente al rango 2-2, entonces la unidad de filtro 559 puede aplicar el filtro 6 a ese píxel o grupo de píxeles y así sucesivamente. El filtro en general puede asumir cualquier tipo de forma o disposición de soporte de filtro. El soporte de filtro se refiere a la forma del filtro con respecto a un píxel dado que se está filtrando, y los

coeficientes de filtro pueden definir la ponderación aplicada a los valores de píxel contiguos de acuerdo con el soporte de filtro. De acuerdo con las técnicas de la presente divulgación, los datos de sintaxis se pueden incluir en el flujo de bits para señalar al descodificador cómo se codificaron los filtros (por ejemplo, cómo se codificaron los coeficientes de filtro), así como los rangos de la métrica de actividad para la que se deben usar diferentes filtros.

[0125] El módulo de predicción 554 recibe la sintaxis de predicción (tal como los vectores de movimiento) desde la unidad de descodificación de entropía 552. Usando la sintaxis de predicción, el módulo de predicción 554 genera los bloques de predicción que se usaron para codificar bloques de vídeo. La unidad de cuantificación inversa 556 realiza la cuantificación inversa, y la unidad de transformada inversa 558 realiza transformadas inversas para cambiar los coeficientes de los bloques de vídeo residuales de nuevo al dominio de píxeles. El sumador 564 combina cada bloque de predicción con la salida del bloque residual correspondiente por la unidad de transformada inversa 558 para reconstruir el bloque de vídeo.

[0126] La unidad de filtro 559 genera los coeficientes de filtro que se van a aplicar para cada entrada de una CU, y a continuación aplica dichos coeficientes de filtro para filtrar los bloques de vídeo reconstruidos de esa CU. El filtrado, por ejemplo, puede comprender un filtrado de eliminación de efecto pixelado adicional que suaviza los bordes y/o elimina los artefactos asociados a los bloques de vídeo, un filtrado de eliminación de ruido para reducir el ruido de cuantificación o cualquier otro tipo de filtrado que pueda mejorar la calidad de la codificación. Los bloques de vídeo filtrados se acumulan en la memoria 562 para reconstruir las tramas descodificados (u otras unidades descodificables) de información de vídeo. Las unidades descodificadas se pueden emitir desde el descodificador de vídeo 560 para su presentación a un usuario, pero también se pueden almacenar para su uso en la descodificación predictiva posterior.

[0127] En el campo de la codificación de vídeo, es común aplicar el filtrado en el codificador y descodificador para potenciar la calidad de una señal de vídeo descodificada. El filtrado se puede aplicar por medio de un filtro posterior, caso en el que la trama filtrada no se usa para la predicción de futuras tramas. De forma alternativa, el filtrado se puede aplicar "en bucle", caso en el que la trama filtrada se puede usar para predecir futuras tramas. Se puede diseñar un filtro deseable minimizando el error entre la señal original y la señal filtrada descodificada. Típicamente, dicho filtrado se ha basado en la aplicación de uno o más filtros a una imagen reconstruida. Por ejemplo, se podría aplicar un filtro de eliminación de efecto pixelado a una imagen reconstruida antes de que la imagen se almacene en la memoria, o se podrían aplicar un filtro de eliminación de efecto pixelado y un filtro adicional a una imagen reconstruida antes de que la imagen se almacene en la memoria.

[0128] De manera similar a la cuantificación de los coeficientes de transformada, también se pueden cuantificar los coeficientes del filtro $h(k,l)$, donde $k = -K, \dots, K$ y $l = -L, \dots, L$. K y L pueden representar valores enteros. Se pueden cuantificar los coeficientes de filtro $h(k,l)$ como:

$$f(k,l) = \text{redondear}(\text{normFact} \cdot h(k,l))$$

donde *normFact* es un factor de normalización y *redondeo* es la operación de redondeo realizada para lograr la cuantificación a una profundidad de bits deseada. La cuantificación de los coeficientes de filtro se puede realizar por la unidad de filtro 349 de la FIG. 3 durante la codificación, y la descuantificación o cuantificación inversa se puede realizar en los coeficientes de filtro descodificados por la unidad de filtro 559 de la FIG. 5. El filtro $h(k,l)$ está destinado a representar de forma genérica cualquier filtro. Por ejemplo, se podría aplicar el filtro $h(k,l)$ a cualquiera de las múltiples entradas. En algunos casos, las múltiples entradas asociadas a un bloque de vídeo utilizarán diferentes filtros, caso en el que se pueden cuantificar y descuantificar múltiples filtros similares a $h(k,l)$ como se describe anteriormente.

[0129] Los coeficientes de filtro cuantificados se codifican y se envían desde el dispositivo de origen asociado al codificador 350 a un dispositivo de destino asociado al descodificador 560 como parte de un flujo de bits codificado. En el ejemplo anterior, el valor de *normFact* suele ser igual a 2^n , aunque se podrían usar otros valores. Los valores más grandes de *normFact* dan lugar a una cuantificación más precisa, de modo que los coeficientes de filtro cuantificados $f(k,l)$ proporcionan un mejor rendimiento. Sin embargo, los valores más grandes de *normFact* pueden producir coeficientes $f(k,l)$ que requieren más bits para señalar al descodificador.

[0130] En el descodificador 560, se pueden aplicar los coeficientes de filtro descodificados $f(k,l)$ a la entrada apropiada. Por ejemplo, si los coeficientes de filtro descodificados se van a aplicar a RI, los coeficientes de filtro se pueden aplicar a la imagen reconstruida posdespixelada $Rl(i,j)$, donde $i=0, \dots, M$ y $j=0, \dots, N$ como sigue:

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

Las variables M, N, K y L pueden representar números enteros. K y L pueden definir un bloque de píxeles que abarca dos dimensiones de -K a K y de -L a L. Los filtros aplicados a otras entradas se pueden aplicar de manera análoga.

[0131] Las técnicas de la presente divulgación pueden mejorar el rendimiento de un filtro posterior o filtro en bucle, y también pueden reducir el número de bits necesarios para señalar los coeficientes de filtro $f(k,l)$. En algunos casos, se señalan una cantidad de filtros posteriores o filtros en bucle diferentes al descodificador para cada serie de bloques de vídeo, por ejemplo, para cada trama, fragmento, parte de una trama, grupo de tramas (GOP) o similares. Para cada filtro, se incluye información adicional en el flujo de bits para identificar las CU, macrobloques y/o píxeles para los que se debe aplicar un filtro dado.

[0132] Las tramas se pueden identificar por número de trama y/o tipo de trama (por ejemplo, tramas I, tramas P o tramas B). Las tramas I se refieren a intratramas que se intrapredicen. Las tramas P se refieren a tramas predictivas que tienen bloques de vídeo predichos en base a una lista de datos (por ejemplo, una trama previa). Las tramas B se refieren a tramas predictivas bidireccionales que se predicen en base a dos listas de datos (por ejemplo, una trama previa y una posterior). Los macrobloques se pueden identificar enumerando los tipos de macrobloques y/o el rango de valores de parámetros de cuantificación (QP) usados para reconstruir el macrobloque.

[0133] Los coeficientes de filtro $f(k,l)$, para cualquier entrada, se pueden codificar usando la predicción de los coeficientes señalados para las CU previas. Para cada entrada de una CU m (por ejemplo, cada trama, fragmento o GOP), el codificador puede codificar y transmitir un conjunto de filtros M:

g_i^m en el que $i=0, \dots, M-1$.

Para cada filtro, el flujo de bits también se puede codificar para identificar la combinación de rangos para dos o más métricas para las que se debe usar el filtro.

[0134] Los coeficientes de filtro se pueden predecir usando coeficientes de filtro reconstruidos usados en una CU previa. Los coeficientes de filtro previos se pueden representar como:

f^n donde $i=0, \dots, N-1$,

En este caso, se puede usar el número de las CU n para identificar uno o más filtros usados para la predicción de los filtros actuales, y se puede enviar el número n al descodificador como parte del flujo de bits codificado. Además, se puede codificar y transmitir la información al descodificador para identificar combinaciones de rangos para dos o más métricas para las que se usa la codificación predictiva.

[0135] La amplitud de los coeficientes de filtro $g(k,l)$ depende de los valores k y l . Normalmente, el coeficiente con la mayor amplitud es el coeficiente $g(0,0)$. Los otros coeficientes que se espera que tengan grandes amplitudes son los coeficientes para los que el valor de k o l es igual a 0. Este fenómeno se puede utilizar para reducir además la cantidad de bits necesarios para señalar los coeficientes. Los valores de índice k y l pueden definir localizaciones dentro de un soporte de filtro conocido.

[0136] Los coeficientes:

$$g_i^m(k,l), i=0, \dots, M-1$$

se pueden codificar para cada trama m usando códigos de longitud variable parametrizados tales como los códigos Golomb o Golomb exponencial definidos de acuerdo con un parámetro p . Al cambiar el valor del parámetro p que define los códigos de longitud variable parametrizados, estos códigos se pueden usar para representar eficazmente una amplia gama de distribuciones de origen. La distribución de los coeficientes $g(k,l)$ (es decir, su probabilidad de tener valores grandes o pequeños) depende de los valores de k y l . Por consiguiente, para incrementar la eficacia de codificación, para cada trama m , se transmite el valor del parámetro p para cada par (k,l) . Se puede usar el parámetro p para la codificación de longitud variable parametrizada cuando se codifican los coeficientes:

$$g_i^m(k,l) \text{ donde } k = -K, \dots, K, l = -L, \dots, L.$$

[0137] Como se describe anteriormente, el descodificador de vídeo 560 representa un ejemplo de un descodificador de vídeo configurado para determinar una primera métrica para un grupo de píxeles dentro de un

bloque de píxeles, determinar una segunda métrica para el grupo de píxeles, determinar un filtro en base a la primera métrica y la segunda métrica y generar una imagen filtrada aplicando el filtro al grupo de píxeles. El decodificador de vídeo 560 también representa un ejemplo de un codificador de vídeo configurado para determinar una primera métrica para un bloque de píxeles, en el que la primera métrica se determina en base a una comparación de un subconjunto de los píxeles en el bloque con otros píxeles en el bloque; determinar una segunda métrica para el bloque de píxeles; determinar un filtro en base a la primera métrica y la segunda métrica; y, generar una imagen filtrada aplicando el filtro al bloque de píxeles.

[0138] Como se describe anteriormente, el decodificador de vídeo 560 también representa un ejemplo de un decodificador de vídeo configurado para determinar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica, en el que cada combinación de rango tiene una identificación (ID) de combinación de rango exclusiva, en el que cada ID de combinación de rango exclusiva corresponde a un valor secuencial para una combinación de rango; asignar las ID de grupo exclusivas a grupos de combinaciones de rango en base a los valores secuenciales para las combinaciones de rango, en el que cada ID de grupo exclusiva corresponde a un valor secuencial para un grupo; y, codificar conjuntos de coeficientes de filtro correspondientes para los filtros en base a las ID de grupo exclusivas. El decodificador de vídeo 560 puede codificar los conjuntos de coeficientes de filtro comprendidos generando los conjuntos de coeficientes de filtro en base a la información recibida en un flujo de bits codificado. El decodificador de vídeo 560 puede generar los conjuntos de coeficientes de filtro usando técnicas de codificación diferencial.

[0139] El decodificador de vídeo 560 también representa un ejemplo de un decodificador de vídeo configurado para asignar una primera combinación de rango a un primer filtro, en el que la primera combinación de rango comprende un primer rango de valores para una primera métrica y un primer rango de valores para una segunda métrica; asignar una segunda combinación de rango a un segundo filtro, en el que la combinación de segundo rango comprende un segundo rango de valores para la primera métrica y un segundo rango de valores para la segunda métrica; asignar una combinación de rango actual a un filtro, en el que la combinación de rango actual comprende el primer rango de valores de la primera métrica y el segundo rango de valores para la segunda métrica. La asignación de la combinación de rango actual al filtro puede incluir asignar la combinación de rango actual al primer filtro en respuesta a la recepción de una primera contraseña, en la que la primera contraseña indica que la combinación de rango actual está asignada al mismo filtro que la primera combinación de rango; asignar la combinación de rango actual al segundo filtro en respuesta a la recepción de una segunda contraseña, en la que la segunda contraseña indica que la combinación de rango actual está asignada al mismo filtro que la segunda combinación; y, asignar la combinación de rango actual a un tercer filtro en respuesta a la recepción de una tercera contraseña, en la que la tercera contraseña identifica ese tercer filtro. El decodificador de vídeo 560 también representa un ejemplo de un decodificador de vídeo configurado para generar una asignación de combinaciones de rango a filtros, en el que una combinación de rango comprende un rango para una primera métrica y un rango para una segunda métrica; asignar una combinación de rango actual a un mismo filtro que una combinación de rango previa en respuesta a la recepción de una primera contraseña que señala que la combinación de rango actual está asignada al mismo filtro que la combinación de rango previa; y, asignar la combinación de rango actual a un filtro identificado por una segunda contraseña en respuesta a la recepción de la segunda contraseña que señala que la combinación de rango actual está asignada a un filtro diferente que la combinación de rango previa.

[0140] Como se ha introducido anteriormente, se pueden usar varios tipos diferentes de métricas junto con las técnicas de filtrado multimétrico descritas en la presente divulgación. Algunas de estas métricas son métricas de actividad que cuantifican la actividad asociada con uno o más bloques de píxeles dentro de los datos de vídeo. Las métricas de actividad pueden comprender métricas de varianza indicativas de la varianza de píxel dentro de un conjunto de píxeles. Como se describirá, algunas de estas métricas de actividad son específicas de la dirección. Por ejemplo, una métrica de actividad horizontal cuantifica la actividad a lo largo de un eje horizontal, una métrica de actividad vertical cuantifica la actividad a lo largo de un eje vertical, una métrica de actividad diagonal cuantifica la actividad a lo largo de un eje diagonal y así sucesivamente.

[0141] Algunas métricas de actividad son no específicas de la dirección. Por ejemplo, un valor laplaciano modificado por suma es una métrica de actividad basada en una ventana bidimensional de píxeles que rodea un píxel actual o un grupo actual de píxeles. Para un píxel actual (i,j) , un valor laplaciano modificado por suma se puede calcular como sigue:

$$\text{var}(i,j) = \sum_{k=-K}^K \sum_{l=-L}^L |2R(i+k,j+l) - R(i+k-1,j+l) - R(i+k+1,j+l)| + \\ |2R(i+k,j+l) - R(i+k,j+l-1) - R(i+k,j+l+1)| \quad (1)$$

donde k representa un valor de una suma de valores de píxel de $-K$ a K y l representa un valor de una suma de $-L$ a L para una ventana bidimensional que abarca de $-K$ a K y $-L$ a L , en el que i y j representan coordenadas de

píxel de los datos de píxel, $Rl(i,j)$ representa un valor de píxel dado en las coordenadas i y j , y $var(i,j)$ es la métrica de actividad (es decir, el valor laplaciano modificado por suma).

[0142] Las técnicas de la presente divulgación también se pueden implementar usando métricas específicas de la dirección para la actividad horizontal, la actividad vertical y la actividad diagonal. Las ecuaciones 2 y 3 muestran ejemplos de cómo se puede calcular la actividad horizontal y la actividad vertical para un píxel actual (x, y) comparando un valor de píxel (Rec), tal como la intensidad, del píxel actual con un valor de píxel de píxeles contiguos.

$$Hor_act(x,y) = R(2*Rec[x][y]-Rec[x+1][y]-Rec[x-1][y]) \quad (2)$$

$$Ver_act(x,y) = R(2*Rec[x][y]-Rec[x][y+1]-Rec[x][y-1]) \quad (3)$$

Como se muestra en la ecuación 2, cuando se determina la actividad horizontal, el píxel actual (x,y) se puede comparar con uno contiguo izquierdo ($x-1, y$) y uno contiguo derecho ($x+1, y$). Como se muestra en la ecuación 3, cuando se determina la actividad vertical, el píxel actual se puede comparar con uno contiguo superior ($x, y+1$) y uno contiguo inferior ($x, y-1$).

[0143] Las ecuaciones 4 y 5 muestran ejemplos de cómo se puede calcular la actividad diagonal para un píxel actual (x, y) comparando un valor de píxel (Rec) del píxel actual con los valores de píxel de los píxeles contiguos.

$$45deg_act(x,y) = R(2*Rec[x][y]-Rec[x+1][y+1]-Rec[x-1][y-1]) \quad (4)$$

$$135deg_act(x,y) = R(2*Rec[x][y]-Rec[x-1][y+1]-Rec[x+1][y-1]) \quad (5)$$

Como se muestra en la ecuación 4, se puede calcular la actividad diagonal, por ejemplo, en la dirección de 45 grados comparando un píxel actual (x, y) con uno contiguo derecho superior ($x+1, y+1$) y uno contiguo izquierdo inferior ($x-1, y-1$). Como se muestra en la ecuación 5, la actividad diagonal también puede estar en la dirección de 135 grados comparando un píxel actual (x, y) con uno contiguo izquierdo superior ($x-1, y+1$) y uno contiguo derecho inferior ($x+1, y-1$).

[0144] Las ecuaciones 2-5 anteriores ilustran cómo se pueden determinar la actividad horizontal, la actividad vertical y la actividad diagonal de píxel en píxel, pero en algunas implementaciones, la actividad horizontal, la actividad vertical y la actividad diagonal se pueden determinar de grupo en grupo, donde un grupo de píxeles es un bloque de píxeles de 2×2 , 4×4 o $M \times N$. En una implementación de este tipo, se puede determinar la actividad horizontal, por ejemplo, comparando los valores de píxel de un grupo actual con los valores de píxel de un grupo izquierdo y un grupo derecho, de manera análoga a la ecuación 2; y, se puede determinar la actividad vertical comparando un grupo actual con un grupo superior y un grupo inferior, de manera análoga a la ecuación 3. Asimismo, se puede determinar la actividad diagonal de 45 grados comparando un grupo actual de píxeles con un grupo contiguo derecho superior y un grupo contiguo izquierdo inferior de manera análoga a la ecuación 4, y se puede determinar la actividad diagonal de 135 grados comparando un grupo actual de píxeles con un grupo contiguo izquierdo superior y un grupo contiguo derecho inferior, de manera análoga a la ecuación 5.

[0145] En algunas implementaciones, se pueden determinar la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados comparando un píxel o grupo de píxeles actual con píxeles o grupos de píxeles contiguos en una sola dirección. Por ejemplo, en lugar de determinar la actividad horizontal en base a la comparación de un píxel actual con uno contiguo izquierdo y uno contiguo derecho, se podría determinar la actividad horizontal en base a solo en uno contiguo izquierdo o solo uno contiguo derecho. Adicionalmente, en algunas implementaciones, se pueden determinar la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados usando promedios o promedios ponderados de áreas de píxeles contiguos en lugar de píxeles contiguos individuales o grupos de píxeles individuales.

[0146] Los valores resultantes de las ecuaciones 2-5 se pueden dividir en un número finito de rangos, tales como 2, 4, 8 o cualquier otro número finito, y se puede asignar una identificación de rango a cada rango. En referencia de nuevo a la FIG. 4A, por ejemplo, el rango 1-1, rango 1-2, rango 2-1, etc. son todos ejemplos de identificaciones de rango. Como un ejemplo, se pueden dividir los valores de actividad horizontal en cuatro rangos, y se podrían asignar ID a los rangos rango 1-1, rango 1-2, rango 1-3 y rango 1-4. Los valores de umbral horizontal (es decir, ThH_1, \dots, ThH_P) pueden determinar dónde comienzan y terminan los rangos. La tabla 2 a continuación muestra el caso genérico de cómo se podrían asignar ID horizontales a los rangos P.

Tabla 2: Índice de métrica de actividad

Condición de hor - act - B	ID horizontal
$\text{Hor_act_B} < \text{ThH}_1$	Rango 2-1
$\text{ThH}_1 \leq \text{Hor_act_B} < \text{ThH}_2$	Rango 2-2
...	...
$\text{ThH}_{P-1} \leq \text{Hor_act_B}$	Rango 2-P

Usando el ejemplo de la tabla 2, si un píxel actual tiene un valor de actividad horizontal mayor que ThH_1 pero menor que ThH_2 , entonces el píxel actual está en el rango 2-2 para la métrica 2. Se pueden asignar los píxeles actuales a rangos verticales con ID verticales, rangos diagonales de 45 grados con ID diagonales de 45 grados y rangos diagonales de 135 grados con ID diagonales de 135 grados, de manera similar a la descrita anteriormente en la tabla 2 para rangos horizontales e ID horizontales.

[0147] Se puede usar cualquiera de la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados como una métrica de acuerdo con las técnicas de filtrado de filtro multimétrico descritas en la presente divulgación. Por ejemplo, haciendo referencia de nuevo a la FIG. 4A, la métrica 1 podría ser una medida de la actividad vertical, y la métrica 2 podría ser una medida de la actividad horizontal. En un ejemplo de este tipo, una unidad de filtro, tal como la unidad de filtro 349 de la FIG. 4A o el filtro 559 de la FIG. 5, puede determinar un filtro para un píxel o grupo de píxeles en base a la actividad horizontal del píxel o grupo de píxeles y la actividad vertical del píxel o grupo de píxeles. Si, por ejemplo, un píxel actual tiene una métrica de actividad horizontal que se encuentra en el rango 2-3 y una métrica de actividad vertical que se encuentra en el rango 1-3, entonces la unidad de filtro filtra el píxel usando el filtro 4. De manera similar, también se pueden usar combinaciones de actividad diagonal de 45 grados y actividad diagonal de 135 grados, actividad diagonal de 45 grados y actividad horizontal, actividad diagonal de 45 grados y actividad vertical, actividad diagonal de 135 grados y actividad horizontal, o actividad diagonal de 135 grados y actividad vertical por una unidad de filtro para seleccionar un filtro para un píxel o grupo de píxeles. En algunas implementaciones, se pueden usar tres o las cuatro de la actividad horizontal, actividad vertical, actividad diagonal de 45 grados y actividad diagonal de 135 grados por una unidad de filtro para seleccionar un filtro de un píxel o grupo de píxeles.

[0148] En las implementaciones descritas anteriormente, se pueden usar la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados como métricas, como la métrica 1 y/o la métrica 2 en la FIG. 4A, por ejemplo. Sin embargo, en algunas implementaciones, la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados podrían no ser métricas por sí mismas, sino que se pueden usar como determinaciones intermedias para determinar una métrica de dirección global. La métrica de dirección en general describe en qué dirección (por ejemplo, sin dirección, horizontal, vertical, diagonal de 45 grados o diagonal de 135 grados) están cambiando más los píxeles.

[0149] En un ejemplo, usando solo la actividad horizontal y la actividad vertical como se describe en las ecuaciones 2 y 3, se podría determinar una dirección para un píxel en base a las siguientes condiciones:

Dirección 1 = horizontal, si $\text{Hor_activity} > k_1 * \text{Ver_activity}$

Dirección 2 = vertical, si $\text{Ver_activity} > k_2 * \text{Hor_activity}$

Dirección 0 = sin dirección, de otro modo.

Las constantes, k_1 y k_2 , se pueden seleccionar de modo que la dirección solo se considere dirección 1 o dirección 2 si la actividad horizontal es sustancialmente mayor que la actividad vertical o la actividad vertical es sustancialmente mayor que la actividad horizontal. Si la actividad horizontal y la actividad vertical son iguales o aproximadamente iguales, entonces la dirección es dirección 0. La dirección 1 en general indica que los valores de píxel están cambiando más en la dirección horizontal que en la dirección vertical, y la dirección 2 indica que los valores de píxel están cambiando más en la dirección vertical que en la dirección horizontal. La dirección 0 indica que el cambio en los valores de píxel en la dirección horizontal es aproximadamente igual al cambio en los valores de píxel en la dirección vertical.

[0150] La métrica de dirección determinada (por ejemplo, dirección 0, dirección 1, dirección 2) se puede usar como métrica en las técnicas de filtrado multimétrico descritas en la presente divulgación. Usando el ejemplo de la FIG. 4A de nuevo, la métrica 1 podría ser una métrica de varianza, tal como un valor laplaciano modificado por suma, mientras que la métrica 2 podría ser una determinación de dirección como se describe anteriormente. Como se describe en referencia a la FIG. 4A, cada una de la dirección 1, dirección 2 y dirección 0 se puede asociar con un rango de la métrica 2, aunque la dirección 1, la dirección 2 y la dirección 0 representan determinaciones finitas en lugar de un espectro de valores.

[0151] Además de usar solo la actividad horizontal y la actividad vertical como se describe anteriormente, las técnicas de la presente divulgación también incluyen usar la actividad diagonal de 45 grados y la actividad diagonal de 135 grados, como se describe en las ecuaciones 4 y 5, para determinar las direcciones, en base a las siguientes condiciones:

Dirección = 1, si $45deg_activity > k1 * 135deg_activity$

Dirección = 2, si $135deg_activity > k2 * 45deg_activity$

Dirección = 0, de otro modo.

Las determinaciones de dirección en base a la actividad diagonal de 45 grados y la actividad diagonal de 135 grados se pueden usar como una métrica con otra métrica, tal como un valor laplaciano modificado por suma, como se describe anteriormente.

[0152] Adicionalmente, también se puede determinar una métrica de dirección, en base a las siguientes condiciones:

Dirección = 1, si $45deg_activity > k1 * 135deg_activity, k2 * Hor_activity, Y k3 * Ver_activity$

Dirección = 2, si $135deg_activity > k4 * 45deg_activity, k5 * Hor_activity, Y k6 * Ver_activity$

Dirección = 3, si $Hor_activity > k7 * Ver_activity, k8 * 135deg_activity, Y k9 * 45deg_activity$

Dirección = 4, si $Ver_activity > k10 * Hor_activity, k11 * 135deg_activity, Y k12 * 45deg_activity$

Dirección = 0, de otro modo.

Como se describe anteriormente, de $k1$ a $k12$ son constantes seleccionadas para la determinación de cuánto mayor que una de la actividad horizontal, actividad vertical, actividad de 45 grados y actividad de 135 grados se necesita que sea en comparación con las otras para seleccionar una determinada dirección. Las determinaciones de dirección en base a la actividad horizontal, la actividad vertical, la actividad diagonal de 45 grados y la actividad diagonal de 135 grados se pueden usar como una métrica con otra métrica, tal como un valor laplaciano modificado por suma, como se describe anteriormente.

[0153] Otra métrica que se puede usar con las técnicas de la presente divulgación incluye una métrica de borde. Una métrica de borde en general cuantifica la actividad que podría ser indicativa de la presencia de un borde en un bloque de píxeles. Un borde se puede producir, por ejemplo, en un bloque de píxeles si ese bloque de píxeles contiene el límite de un objeto dentro de una imagen. Un ejemplo de detección de bordes incluye usar los cuatro píxeles contiguos a un píxel actual (por ejemplo, izquierdo, derecho, superior, inferior) o usar los ocho píxeles contiguos al píxel actual (izquierdo, derecho, superior, inferior, superior derecho, superior izquierdo, inferior derecho, inferior izquierdo). Adicionalmente, la detección del tipo de borde puede incluir usar dos píxeles contiguos, tales como superior e inferior, izquierdo y derecho, superior izquierdo e inferior derecho, o superior derecho e inferior izquierdo.

[0154] El pseudocódigo a continuación muestra ejemplos de cómo se puede calcular la información de borde para un píxel actual (x, y) comparando un valor de píxel (Rec), tal como la intensidad, del píxel actual con los valores de píxel de los píxeles contiguos (es decir, 4/8 píxeles).

[0155] Una variable EdgeType se inicia en 0. Cada vez que una declaración es verdadera, la variable EdgeType se incrementa en 1 (como se muestra en el pseudocódigo por EdgeType++) o bien se disminuye en 1 (como se muestra en el pseudocódigo por EdgeType--). Rec[x][y] se refiere a un valor de píxel, tal como la intensidad de píxel, del píxel localizado en (x, y). La primera agrupación de declaraciones "si (if)" es para comparar el píxel actual con los contiguos superior, inferior, izquierdo y derecho. La segunda agrupación de declaraciones "si" es para comparar el píxel actual con los contiguos superior izquierdo, superior derecho, inferior izquierdo e inferior derecho. Las técnicas de la presente divulgación se pueden implementar usando un grupo o bien ambos grupos.

```

EdgeType = 0;
if (Rec[x][y] > Rec[x-1][y]) EdgeType ++;
if (Rec[x][y] < Rec[x-1][y]) EdgeType --;
if (Rec[x][y] > Rec[x+1][y]) EdgeType ++;
if (Rec[x][y] < Rec[x+1][y]) EdgeType --;
if (Rec[x][y] > Rec[x][y-1]) EdgeType ++;
if (Rec[x][y] < Rec[x][y-1]) EdgeType --;
if (Rec[x][y] > Rec[x][y+1]) EdgeType ++;
if (Rec[x][y] < Rec[x][y+1]) EdgeType --;

if (Rec[x][y] > Rec[x-1][y-1]) EdgeType ++;
if (Rec[x][y] < Rec[x-1][y-1]) EdgeType --;
if (Rec[x][y] > Rec[x+1][y-1]) EdgeType ++;
if (Rec[x][y] < Rec[x+1][y-1]) EdgeType --;
if (Rec[x][y] > Rec[x-1][y+1]) EdgeType ++;
if (Rec[x][y] < Rec[x-1][y+1]) EdgeType --;
if (Rec[x][y] > Rec[x+1][y+1]) EdgeType ++;
if (Rec[x][y] < Rec[x+1][y+1]) EdgeType --;

```

[0156] Si un píxel actual es un máximo local, entonces el valor de píxel del píxel será mayor que todos sus contiguos y tendrá un tipo de borde de 4 si usa los cuatro contiguos o un tipo de borde de 8 si usa los ocho contiguos. Si un píxel actual es mínimo local, entonces el valor de píxel del píxel será menor que todos sus contiguos y tendrá un tipo de borde de -4 si usa los cuatro contiguos o un tipo de borde de -8 si usa los ocho contiguos. Por tanto, el uso de las técnicas de ejemplo descritas anteriormente para determinar un tipo de borde entre -4 y 4 o -8 y 8 se puede usar en la determinación de un filtro. Se pueden asignar los valores determinados para el tipo de borde (es decir, valores de -4 a 4 o valores de -8 a 8) a rangos de una métrica, tal como la métrica 1 o la métrica 2 de la FIG. 4A. En algunas implementaciones, se podrían asignar los valores absolutos de la determinación del tipo de borde a rangos, de modo que se asignaría un tipo de borde de -3 y 3, por ejemplo, al mismo filtro.

[0157] Los cálculos de las diversas métricas descritas en la presente divulgación solo pretenden ser ejemplos y no son exhaustivos. Por ejemplo, se pueden determinar las métricas usando ventanas o líneas de píxeles que incluyen más píxeles contiguos que los descritos en la presente divulgación.

[0158] Adicionalmente, en algunas implementaciones, se pueden calcular las métricas descritas en la presente divulgación usando submuestreo de los píxeles en una línea o ventana particular. Por ejemplo, para calcular una métrica de actividad de bloque para un bloque de píxeles 4x4, las métricas de actividad y dirección se pueden calcular como sigue:

- Métrica de dirección

- $Ver_act(i,j) = \text{abs}(X(i,j) - X(i,j-1) - X(i,j+1))$
- $Hor_act(i,j) = \text{abs}(X(i,j) - X(i-1,j) - X(i+1,j))$
- $H_B = \sum_{i=0,2} \sum_{j=0,2} Hor_act(i,j)$
- $V_B = \sum_{i=0,2} \sum_{j=0,2} Ver_act(i,j)$
- Dirección = 0, 1 ($H_B > k_1 \cdot V_B$), 2 ($V_B > k_2 \cdot H_B$)

- Métrica de actividad

- $L_B = H_B + V_B$
- 5 clases (0, 1, 2, 3, 4)

- Métrica

- Combinación de actividad y dirección (por ejemplo, 15 o 16 combinaciones como se explica anteriormente en el ejemplo de la FIG. 4B)

[0159] $Hor_act(i,j)$ se refiere en general a la actividad horizontal del píxel actual (i, j), y $Ver_act(i,j)$ se refiere en general a la actividad vertical del píxel actual (i, j). $X(i,j)$ se refiere en general a un valor de píxel del píxel (i, j). H_B se refiere a la actividad horizontal del bloque 4x4, que en este ejemplo se determina en base a una suma de la actividad horizontal para los píxeles (0, 0), (0, 2), (2, 0) y (2, 2). V_B se refiere a la actividad vertical del bloque 4x4, que en este ejemplo se determina en base a una suma de la actividad vertical para los píxeles (0, 0), (0, 2), (2, 0) y (2, 2). "<<1" representa una operación de multiplicar por dos. Como se explica anteriormente, en base a los

valores de H_B y V_B , se puede determinar una dirección. Usando el ejemplo anterior, si el valor de H_B es más de k veces el valor de V_B , entonces se puede determinar que la dirección es la dirección 1 (es decir, horizontal), que podría corresponder a una actividad más horizontal que la actividad vertical. Si el valor de V_B es más de k veces el valor de H_B , entonces se puede determinar que la dirección es la dirección 2 (es decir, vertical), que podría corresponder a una actividad más vertical que la actividad horizontal. De otro modo, se puede determinar que la dirección es la dirección 0 (es decir, sin dirección), que significa que no domina ni la actividad horizontal ni la vertical. Las etiquetas para las diversas direcciones y las proporciones usadas para determinar las direcciones constituyen simplemente un ejemplo, ya que también se pueden usar otras etiquetas y proporciones.

[0160] La actividad (L_B) para el bloque 4x4 se puede determinar como una suma de la actividad horizontal y vertical. El valor de L_B se puede clasificar en un rango, como se describe anteriormente. Este ejemplo particular muestra cinco rangos, aunque se pueden usar más o menos rangos de forma similar. En base a la combinación de actividad y dirección, se puede seleccionar un filtro para el bloque de píxeles 4x4. Como se describe anteriormente, se puede seleccionar un filtro en base a una asignación bidimensional de la actividad y la dirección a los filtros, como se describe en referencia a las FIGS. 4A y 4B, o se pueden combinar la actividad y la dirección en una única métrica, y se puede usar esa única métrica para seleccionar un filtro.

[0161] La FIG. 6A representa un bloque de píxeles 4x4. Usando las técnicas de submuestreo descritas anteriormente, solo se usan cuatro de los dieciséis píxeles. Los cuatro píxeles son el píxel (0, 0) que está etiquetado como píxel 601, el píxel (2, 0) que está etiquetado como píxel 602, el píxel (0, 2) que está etiquetado como píxel 603 y el píxel (2, 2) que está etiquetado como píxel 604. La actividad horizontal del píxel 601 (es decir, $hor_act(0, 0)$), por ejemplo, se determina en base a un píxel contiguo izquierdo y a un píxel contiguo derecho. El píxel contiguo derecho está etiquetado como píxel 605. El píxel contiguo izquierdo está localizado en un bloque diferente al bloque 4x4 y no se muestra en la FIG. 6A. La actividad vertical del píxel 602 (es decir, $ver_act(2, 0)$), por ejemplo, se determina en base a un píxel contiguo superior y a un píxel contiguo inferior. El píxel contiguo inferior está etiquetado como píxel 606 y el píxel contiguo superior está localizado en un bloque diferente al bloque 4x4 y no se muestra en la FIG. 6A.

[0162] En general, usando las mismas técnicas descritas anteriormente, también se puede calcular una métrica de actividad de bloque usando un subconjunto diferente de píxeles como sigue:

- Métrica de dirección

- $Ver_act(i,j) = abs(X(i,j) - X(i,j-1) - X(i,j+1))$
- $Hor_act(i,j) = abs(X(i,j) - X(i-1,j) - X(i+1,j))$
- $H_B = \sum_{i=1,2} \sum_{j=1,2} H(i,j)$
- $V_B = \sum_{i=1,2} \sum_{j=1,2} V(i,j)$
- Dirección = 0, 1 ($H > k1 * V$), 2 ($V > k2 * H$)

- Métrica de actividad

$$L_B = H_B + V_B$$

- 5 clases (0, 1, 2, 3, 4)

- Métrica

- Combinación de actividad y dirección (por ejemplo, 15 o 16 combinaciones como se explica anteriormente en el ejemplo de la FIG. 4B)

[0163] Este subconjunto diferente de píxeles para calcular H_B y V_B incluye los píxeles (1, 1), (2, 1), (1, 2) y (2, 2), mostrados en la FIG. 6B como los píxeles 611, 612, 613 y 614, respectivamente. Como se puede ver en la FIG. 6B, todos los píxeles contiguos superiores, píxeles contiguos inferiores, píxeles contiguos derechos y píxeles contiguos izquierdos para los píxeles 611, 612, 613 y 614 están localizados dentro del bloque 4x4. En el ejemplo de la FIG. 6B, los píxeles 611, 612, 613 y 614 están todos localizados en el interior del bloque a diferencia de estar localizados en el límite del bloque. Los píxeles 601, 602, 603 y 605 en la FIG. 6A y los píxeles 621, 624, 625 y 628 en la FIG. 6C son ejemplos de píxeles localizados en el límite del bloque. En otras implementaciones, se pueden elegir diferentes subconjuntos adicionales de píxeles. Por ejemplo, se pueden seleccionar los subconjuntos de modo que los píxeles contiguos superiores e inferiores para los píxeles del subconjunto estén dentro del bloque 4x4, pero algunos píxeles contiguos izquierdos y derechos estén en bloques contiguos. También se pueden seleccionar los subconjuntos de modo que los píxeles contiguos izquierdos y derechos para los píxeles del subconjunto estén dentro del bloque 4x4, pero algunos píxeles contiguos superiores e inferiores estén en bloques contiguos.

[0164] En general, usando las mismas técnicas descritas anteriormente, también se puede calcular una métrica de actividad de bloque usando un subconjunto de ocho píxeles como sigue:

- Métrica de dirección

- $Ver_act(i,j) = \text{abs}(X(i,j) - X(i,j-1) - X(i,j+1))$
- $Hor_act(i,j) = \text{abs}(X(i,j) - X(i-1,j) - X(i+1,j))$
- $H_B = \sum_{i=0,1,2,3} \sum_{j=1,2} H(i,j)$
- $V_B = \sum_{i=0,1,2,3} \sum_{j=1,2} V(i,j)$
- Dirección = 0, 1 ($H > k_1 \cdot V$), 2 ($V > k_2 \cdot H$)

- Métrica de actividad

- $L_B = H_B + V_B$
- 5 clases (0, 1, 2, 3, 4)

- Métrica

- Combinación de actividad y dirección (por ejemplo, 15 o 16 combinaciones como se explica anteriormente en el ejemplo de la FIG. 4B)

[0165] Este subconjunto diferente de ocho píxeles para calcular H_B y V_B incluye píxeles (0, 1), (1, 1), (2, 1), (3, 1), (0, 2), (1, 2), (2, 2) y (3, 2), mostrados en la FIG. 6C como píxeles 621, 622, 623 y 624, 625, 626, 627 y 628 respectivamente. Como se puede ver en la FIG. 6C, todos los píxeles contiguos superiores y los píxeles contiguos inferiores para los píxeles 621, 622, 623 y 624, 625, 626, 627 y 628 están localizados dentro del bloque 4x4, aunque los píxeles 621 y 625 tienen cada uno píxeles contiguos izquierdos en un bloque contiguo izquierdo y los píxeles 624 y 628 tienen cada uno píxeles contiguos derechos en un bloque contiguo derecho. Esta selección particular de píxeles puede reducir la complejidad del codificador y/o decodificador evitando la necesidad de una memoria intermedia de línea para almacenar valores de píxel de bloques contiguos superiores y/o inferiores. Debido al orden de exploración de trama de izquierda a derecha, de arriba a abajo, las memorias intermedias de línea para los valores de píxel de los bloques contiguos superior e inferior a menudo necesitan almacenar valores de píxel para toda la línea superior o inferior, que en el caso del vídeo 1080P, por ejemplo, podría tener 1920 píxeles. Sin embargo, las memorias intermedias de línea para los bloques contiguos izquierdo y derecho a menudo solo necesitan almacenar los valores de píxel para una LCU o un par de LCU, que podrían tener solo 64 o 128 píxeles, por ejemplo. Por tanto, las memorias intermedias de línea para los valores de píxel de bloques contiguos superior e inferior pueden necesitar ser significativamente más grandes que las memorias intermedias de línea usadas para los valores de píxel de los bloques contiguos izquierdo y derecho. La selección de píxeles mostrada en la FIG. 6C puede evitar el uso de memorias intermedias de línea para los valores de píxel del bloque contiguo superior e inferior, reduciendo por tanto la complejidad de la codificación.

[0166] Los ejemplos de las FIGS. 6A-6C son simplemente técnicas introducidas de la presente divulgación. Se contempla que estas técnicas se puedan extender a bloques diferentes a solo 4x4 y que se puedan seleccionar diferentes subconjuntos de píxeles.

[0167] Cuando se calcula una métrica de actividad de bloque, en lugar de píxeles originales, se pueden usar píxeles cuantificados (es decir, $X(i,j) \gg N$) para reducir la complejidad de las operaciones, tales como las operaciones de suma. Adicionalmente, se pueden basar los cálculos en la diferencia absoluta en lugar de basados en laplaciana. Por ejemplo, cuando se calcula $Hor_act(i,j)$ o $Ver_act(i,j)$, se pueden usar diferencias absolutas en lugar de los valores laplacianos, como sigue:

- Métrica de dirección

- $Ver_act(i,j) = \text{abs}(X(i,j) - X(i,j-1))$
- $Hor_act(i,j) = \text{abs}(X(i,j) - X(i-1,j))$
- $H_B = \sum_{i=0,1,2} \sum_{j=0,1,2} H(i,j)$
- $V_B = \sum_{i=0,1,2} \sum_{j=0,1,2} V(i,j)$
- Dirección = 0, 1 ($H > 2V$), 2 ($V > 2H$)

- Métrica de actividad

- $L_B = H_B + V_B$
- 5 clases (0, 1, 2, 3, 4)

- Métrica

- Actividad + dirección (por ejemplo, 15 o 16 combinaciones como se explica anteriormente en el ejemplo de la FIG. 4B)

[0168] La presente divulgación ha descrito técnicas de submuestreo con referencia a un grupo limitado de métricas específicas. Sin embargo, se contempla que estas técnicas de submuestreo en general sean aplicables a otras métricas, tales como las otras métricas analizadas en la presente divulgación, que se pueden usar con propósitos de determinación de un filtro. Adicionalmente, aunque las técnicas de submuestreo de la presente divulgación se han descrito con referencia a bloques de píxeles 4x4, las técnicas también pueden ser aplicables a bloques de otros tamaños.

[0169] La FIG. 7 es un diagrama de flujo que ilustra una técnica de codificación de vídeo consecuente con la presente divulgación. Las técnicas descritas en la FIG. 7 se pueden realizar por la unidad de filtro de un codificador de vídeo o un decodificador de vídeo, tal como la unidad de filtro 349 del codificador de vídeo 350 o la unidad de filtro 559 del decodificador de vídeo 560. La unidad de filtro determina una primera métrica para un grupo de píxeles dentro de un bloque de píxeles (710). La primera métrica, por ejemplo, puede ser una métrica de actividad, tal como un valor laplaciano modificado por suma, o la primera métrica puede ser una métrica de dirección. Se puede determinar la primera métrica, por ejemplo, en base a una comparación del conjunto de píxeles en el bloque, o en base a un subconjunto de los píxeles en el bloque, con otros píxeles en el bloque. La unidad de filtro determina además una segunda métrica para el bloque (720). La segunda métrica, por ejemplo, puede ser una métrica de dirección que se determina en base a la comparación de una medida de actividad horizontal con una medida de actividad vertical. En base a la primera métrica y la segunda métrica, la unidad de filtro determina un filtro (730). La unidad de filtro genera una imagen filtrada aplicando el filtro al bloque (740). Como se analiza anteriormente, en algunas implementaciones, el bloque puede ser un bloque de píxeles 2x2, 4x4 o MxN, usado para determinar la primera métrica o la segunda métrica. En algunas implementaciones, la primera métrica puede ser una métrica de actividad horizontal mientras que la segunda métrica es una métrica de actividad vertical, o la primera métrica puede ser una métrica de borde mientras que la segunda métrica es una métrica de dirección.

[0170] La FIG. 8A es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 8A se pueden realizar por la unidad de filtro de un decodificador de vídeo, tal como la unidad de filtro 559 del decodificador de vídeo 560. La unidad de filtro 559 asigna una primera combinación de rango a un primer filtro (810A). La primera combinación de rango es la combinación de un primer rango de valores para una primera métrica y un primer rango de valores para una segunda métrica. La primera métrica, por ejemplo, puede ser un valor laplaciano modificado por suma y la segunda métrica puede ser una métrica de dirección, aunque también se pueden usar otras métricas. La unidad de filtro 559 asigna una segunda combinación de rango a un segundo filtro (820A). La segunda combinación de rango es una combinación de un segundo rango de valores para la primera métrica y un segundo rango de valores para la segunda métrica. A continuación, la unidad de filtro 559 asigna una combinación de rango actual a un filtro en base a una contraseña recibida. La combinación de rango actual incluye el primer rango de valores de la primera métrica y el segundo rango de valores para la segunda métrica. Si la contraseña es una primera contraseña (830A, sí), entonces la unidad de filtro 559 asigna la combinación de rango actual al primer filtro (840A). La primera contraseña indica que la combinación de rango actual está asignada al mismo filtro que la primera combinación de rango. Si la contraseña es una segunda contraseña (850A, sí), la unidad de filtro 559 asigna la combinación de rango actual al segundo filtro (860A). La segunda contraseña indica que la combinación de rango actual está asignada al mismo filtro que la segunda combinación. Si la contraseña no es una primera contraseña ni una segunda contraseña (850A, no), entonces la unidad de filtro 559 asigna la combinación de rango actual a un tercer filtro (870A). Si en respuesta a la recepción de una tercera contraseña, en la que la tercera contraseña identifica ese tercer filtro. En el ejemplo de la FIG. 8A, la primera contraseña y la segunda contraseña pueden incluir cada una menos bits que la tercera contraseña.

[0171] La FIG. 8B es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 8B se pueden realizar por la unidad de filtro de un decodificador de vídeo, tal como la unidad de filtro 559 del decodificador de vídeo 560. La unidad de filtro 559 genera una asignación de combinaciones de rango a los filtros (810B). Cada combinación de rango, por ejemplo, puede incluir un rango para una primera métrica y un rango para una segunda métrica. En respuesta a la recepción de una primera contraseña que señala que una combinación de rango actual se asigna a un mismo filtro que una combinación de rango previa (820B, sí), la unidad de filtro 559 asigna la combinación de rango actual al mismo filtro que la combinación de rango previa (830B). En respuesta a la recepción de una segunda contraseña que señala que la combinación de rango actual se asigna a un filtro diferente a la combinación de rango previa (820B, no), la unidad de filtro 559 asigna la combinación de rango actual a un nuevo filtro (840B). Como se describe anteriormente, la combinación de rango actual se puede determinar en base a un orden de transmisión conocido. En algunos ejemplos, el nuevo filtro se puede identificar en base a la segunda contraseña, mientras que en otros ejemplos, el nuevo filtro se podría determinar en base al orden en que se señalan los coeficientes de filtro.

[0172] La FIG. 9A es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 9A se pueden realizar por la unidad de filtro de un codificador de vídeo, tal como la unidad de filtro 349 del codificador de vídeo 350. La unidad de filtro 349 determina una asignación de combinaciones de rango a los filtros (910A). Cada combinación de rango incluye un rango de valores para una primera métrica y un rango de valores para una segunda métrica. Para una combinación de rango actual, si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que

comprende el mismo rango de valores para la primera métrica (920A, sí), entonces la unidad de filtro 349 genera una primera contraseña (930A). Si la combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que comprende el mismo rango de valores para la segunda métrica (940A, sí), entonces la unidad de filtro 349 genera una segunda contraseña (950A). Si la combinación de rango actual no se asigna a la combinación de rango previa que comprende el mismo rango de valores para la primera métrica o bien a la combinación de rango previa que comprende el mismo rango de valores para la segunda métrica (950A, no), entonces la unidad de filtro 349 genera una tercera contraseña (960A). La tercera contraseña puede identificar un filtro asignado a la combinación de rango actual.

[0173] La FIG. 9B es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 9BA se pueden realizar por la unidad de filtro de un codificador de vídeo, tal como la unidad de filtro 349 del codificador de vídeo 350. La unidad de filtro 349 determina una asignación de combinaciones de rango a los filtros (910B). Cada combinación de rango, por ejemplo, puede incluir un rango para una primera métrica y un rango para una segunda métrica. Cuando una combinación de rango actual que se está codificando tiene el mismo filtro que una combinación de rango previamente codificada (920B, sí), la unidad de filtro 349 puede generar una primera contraseña para señalar que la combinación de rango actual está asignada al mismo filtro que una combinación de rango previa (930B). Cuando una combinación de rango actual que se está codificando no tiene el mismo filtro que una combinación de rango codificada previamente (920B, no), la unidad de filtro 349 puede generar una segunda contraseña (940B). La segunda contraseña puede identificar el filtro asignado a la combinación de rango actual. Como se describe anteriormente, la combinación de rango actual se puede determinar en base a un orden de transmisión conocido. En el ejemplo de la FIG. 9B, la primera contraseña puede incluir menos bits que la segunda contraseña.

[0174] En los ejemplos de las FIGS. 8A y 8B y las FIGS. 9A y 9B, los términos "primera contraseña", "segunda contraseña" y "tercera contraseña" se usan para diferenciar entre diferentes contraseñas y no tienen la intención de implicar un orden secuencial de contraseñas.

[0175] La FIG. 10 es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 10 se pueden realizar por la unidad de filtro de un codificador de vídeo, tales como la unidad de filtro 349 del codificador de vídeo 350, o la unidad de filtro de un descodificador de vídeo, tal como la unidad de filtro 559. La unidad de filtro determina una asignación de combinaciones de rango a los filtros (1010). Las combinaciones de rango incluyen un rango para una primera métrica y un rango para una segunda métrica. La unidad de filtro determina una identificación (ID) de combinación de rango exclusiva para cada combinación de rango (1020). Las ID de combinación de rango exclusivas corresponden a valores secuenciales. La unidad de filtro asigna una primera ID de grupo exclusiva a un primer grupo de combinaciones de rango en base al valor secuencial de una ID de combinación de rango de al menos una combinación de rango en el primer grupo de combinaciones de rango (1030). Los grupos de combinaciones de rango incluyen combinaciones de rango asignadas al mismo filtro, las ID de grupo exclusivas corresponden a un conjunto de valores secuenciales. La unidad de filtro codifica un primer conjunto de coeficientes de filtro correspondientes al mismo filtro en base al valor secuencial de la primera ID de filtro exclusiva (1040). En el caso del codificador de vídeo, la codificación del primer conjunto de coeficientes de filtro puede incluir, por ejemplo, señalar los coeficientes de filtro en un flujo de bits codificado usando técnicas de codificación diferencial. En el caso de un descodificador de vídeo, la codificación del primer conjunto de coeficientes de filtro puede incluir reconstruir los coeficientes de filtro en base a la información recibida en un flujo de bits codificado.

[0176] La FIG. 11 es un diagrama de flujo que ilustra técnicas de codificación de vídeo consecuentes con la presente divulgación. Las técnicas descritas en la FIG. 11 se pueden realizar por la unidad de filtro de un codificador de vídeo, tales como la unidad de filtro 349 del codificador de vídeo 350, o la unidad de filtro de un descodificador de vídeo, tal como la unidad de filtro 559. La unidad de filtro determina una asignación de combinaciones de rango a los filtros (1110). Las combinaciones de rango pueden incluir un rango para una primera métrica y un rango para una segunda métrica. Cada combinación de rango puede tener una identificación (ID) de combinación de rango exclusiva, y cada ID de combinación de rango exclusiva puede corresponder a un valor secuencial para la combinación de rango. La unidad de filtro puede asignar una ID de grupo exclusiva a cada grupo de combinaciones de rango (1120). La unidad de filtro puede asignar las ID de grupo exclusivas, por ejemplo, en base a los valores secuenciales de las combinaciones de rango. Un grupo de combinaciones de rango puede incluir combinaciones de rango asignadas a un filtro común, y las ID de grupo exclusivas pueden corresponder a un conjunto de valores secuenciales. La unidad de filtro puede codificar conjuntos de coeficientes de filtro para los filtros en base a las ID de grupo exclusivas (1140).

[0177] En el ejemplo de la FIG. 11, la unidad de filtro puede asignar las ID de grupo exclusivas, por ejemplo, asignando una ID de grupo exclusiva correspondiente a un valor secuencial inferior de las ID de grupo exclusivas a un grupo de combinaciones de rango que comprende una combinación de rango con una ID de combinación de rango correspondiente a un valor secuencial inferior de las ID de combinación de rango. En otro ejemplo, la unidad de filtro puede asignar la ID de grupo exclusiva correspondiente a un valor secuencial superior de las ID de grupo exclusivas a un grupo de combinaciones de rango que comprende una combinación de rango con una ID de combinación de rango correspondiente a un valor secuencial superior de las ID combinación de rango.

[0178] En los casos donde la unidad de filtro es parte de un descodificador de vídeo, la unidad de filtro puede codificar los conjuntos de coeficientes de filtro generando los conjuntos de coeficientes de filtro en base a la información recibida en un flujo de bits codificado. La unidad de filtro puede, por ejemplo, generar los conjuntos de coeficientes de filtro usando técnicas de codificación diferencial. En los casos donde la unidad de filtro es parte de un codificador de vídeo, la unidad de filtro puede codificar los conjuntos de coeficientes de filtro señalando los conjuntos de coeficientes de filtro en un flujo de bits codificado en un orden seleccionado en base a los valores secuenciales de las ID de grupo exclusivas. La unidad de filtro, por ejemplo, puede señalar los conjuntos de coeficientes de filtro usando técnicas de codificación diferencial.

[0179] La divulgación anterior se ha simplificado hasta cierto punto para transmitir detalles. Por ejemplo, la divulgación en general describe conjuntos de filtros que se señalan por trama o por fragmento, pero también se pueden señalar los conjuntos de filtros por secuencia, por grupo de imágenes, por grupo de fragmentos, por CU, por LCU u otra base de este tipo. En general, se pueden señalar los filtros para cualquier agrupación de una o más CU. Además, en la implementación, puede haber numerosos filtros por entrada por CU, numerosos coeficientes por filtro y numerosos niveles diferentes de variación con cada uno de los filtros que se están definiendo para un rango diferente de varianza. Por ejemplo, en algunos casos puede haber dieciséis o más filtros definidos para cada entrada de una CU y dieciséis rangos de varianza diferentes correspondientes a cada filtro. Además, cuando la presente divulgación describe la transmisión de información del filtro, no se debe suponer que toda la información del filtro se transmite al mismo nivel de codificación. Por ejemplo, en algunas implementaciones, parte de la información del filtro, tal como la sintaxis de descripción de filtro, se puede señalar de trama en trama o de fragmento en fragmento, mientras que otra información del filtro, tales como los coeficientes de filtro, se señala de LCU en LCU. La sintaxis en otros niveles de la jerarquía de codificación, tal como el nivel de secuencia, el nivel de GOP u otros niveles, también se podría definir para transmitir parte o la totalidad de dicha información del filtro.

[0180] Cada uno de los filtros para cada entrada puede incluir muchos coeficientes. En un ejemplo, los filtros comprenden filtros bidimensionales con 81 coeficientes diferentes definidos para un soporte de filtro que se extiende en dos dimensiones. Sin embargo, el número de coeficientes de filtro que se señalan para cada filtro puede ser menor a 81 en algunos casos. Se puede imponer la simetría del coeficiente, por ejemplo, de modo que los coeficientes de filtro en una dimensión o cuadrante puedan corresponder a valores invertidos o simétricos en relación con los coeficientes en otras dimensiones o cuadrantes. La simetría del coeficiente puede permitir que 81 coeficientes diferentes se representen por menos coeficientes, caso en el que el codificador y el descodificador pueden suponer que los valores invertidos o reflejados de los coeficientes definen otros coeficientes. Por ejemplo, los coeficientes (5, -2, 10, 10, -2, 5) se pueden codificar y señalar como el subconjunto de coeficientes (5, -2, 10). En este caso, el descodificador puede saber que estos tres coeficientes definen el conjunto simétrico más grande de coeficientes (5, -2, 10, 10, -2, 5).

[0181] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un microteléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). Se ha descrito cualquier componente, módulo o unidad proporcionado para enfatizar aspectos funcionales y no necesariamente requieren su realización por diferentes unidades de hardware.

[0182] En consecuencia, las técnicas descritas en el presente documento se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en hardware, cualquier rasgo característico descrito como módulos, unidades o componentes se puede implementar conjuntamente en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperativos. Si se implementan en software, las técnicas se pueden realizar, al menos en parte, mediante un medio legible por ordenador que comprende instrucciones que, cuando se ejecutan en un procesador, realizan uno o más de los procedimientos descritos anteriormente. El medio legible por ordenador puede comprender un medio de almacenamiento legible por ordenador y puede formar parte de un producto de programa informático, que puede incluir materiales de empaquetado. El medio de almacenamiento legible por ordenador puede comprender memoria de acceso aleatorio (RAM) tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de solo lectura programable y borrrable eléctricamente (EEPROM), memoria flash, medios de almacenamiento de datos magnéticos u ópticos y similares. Las técnicas se pueden realizar adicionalmente, o de forma alternativa, al menos en parte por un medio de comunicación legible por ordenador que lleve o comunique un código en forma de instrucciones o estructuras de datos y al que se pueda acceder, leer y/o ejecutar por un ordenador.

[0183] El código se puede ejecutar por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables *in situ* (FPGA) u otro circuito lógico integrado o discreto equivalente. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de software o módulos de hardware dedicados configurados para la codificación y la descodificación, o incorporarse en un códec de vídeo combinado. Además, las técnicas se podrían

implementar por completo en uno o más circuitos o elementos lógicos.

[0184] Se han descrito diversos aspectos de la divulgación. Estos y otros aspectos están dentro del alcance de las siguientes reivindicaciones.

5

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo en un esquema de filtro en bucle adaptativo basado en árbol cuaternario, QALF, con múltiples filtros, comprendiendo el procedimiento:

generar una asignación de combinaciones de rango a dichos múltiples filtros,

en el que una combinación de rango comprende un rango de valores para una primera métrica indicativa de varianza de píxel y un rango de valores para una segunda métrica indicativa de varianza de píxel en una dirección específica,

en el que la primera métrica se determina en base a valores de píxel de un bloque de píxeles,

en el que la segunda métrica se determinada en base a los valores de píxel de dicho bloque de píxeles, y

en el que la segunda métrica es diferente de la primera métrica,

en el que la asignación de combinaciones de rango a dichos múltiples filtros se genera en base a una primera contraseña y una segunda contraseña recibidas desde un codificador,

en el que algunas de las combinaciones de rango se asignan a un mismo filtro; en respuesta a la recepción de la primera contraseña, en el que la primera contraseña señala que una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que comprende el mismo rango de valores para la primera métrica, asignar la combinación de rango actual al mismo filtro,

en el que la combinación de rango actual se determina en base a un orden de transmisión conocido de las combinaciones de rango;

en respuesta a la recepción de la segunda contraseña, en el que la segunda contraseña señala que la combinación de rango actual se asigna a un filtro diferente al de la combinación de rango previa, asignar la combinación de rango actual a un nuevo filtro, en el que el nuevo filtro se determina en base a un orden de transmisión en el que se señalan los coeficientes de filtro.

2. El procedimiento de la reivindicación 1,

en el que la métrica de dirección se determina en base a la comparación de una medida de actividad horizontal con una medida de actividad vertical, en el que la métrica de actividad cuantifica la actividad asociada al bloque de píxeles.

3. El procedimiento de la reivindicación 2, en el que la medida de actividad horizontal se determina en base a la comparación de un valor de píxel de un píxel actual con un valor de píxel de un píxel vecino izquierdo y un valor de píxel de un píxel vecino derecho, y en el que la medida de la actividad vertical se determina en base a la comparación de un valor de píxel de un píxel actual con un valor de píxel de un píxel vecino superior y un valor de píxel de un píxel vecino inferior.

4. El procedimiento de la reivindicación 2, en el que la primera métrica cuantifica la actividad asociada al bloque de píxeles, en el que la primera métrica se determina en base a una suma de la medida de actividad horizontal y la medida de actividad vertical.

5. Un aparato que comprende medios para llevar a cabo el procedimiento de una cualquiera de las reivindicaciones 1 a 4.

6. Un procedimiento de codificación de datos de vídeo en un esquema de filtro en bucle adaptativo basado en árbol cuaternario, QALF, con múltiples filtros, comprendiendo el procedimiento:

determinar una asignación de combinaciones de rango a dichos múltiples filtros, en el que una combinación de rango comprende un rango de valores para una primera métrica indicativa de varianza de píxel y un rango de valores para una segunda métrica indicativa de varianza de píxel en una dirección específica, en el que algunas de las combinaciones de rango se asignan a un mismo filtro;

en el que la primera métrica se determina en base a valores de píxel de un bloque de píxeles,

en el que la segunda métrica se determinada en base a los valores de píxel de dicho bloque de píxeles, y

en el que la segunda métrica es diferente de la primera métrica;

generar una primera contraseña si una combinación de rango actual se asigna al mismo filtro que una combinación de rango previa que comprende el mismo rango de valores para la primera métrica, en el que la combinación de rango actual se determina en base a un orden de transmisión conocido de las combinaciones de rango;

generar una segunda contraseña si la combinación de rango actual se asigna a un filtro diferente a la combinación de rango previa, en el que la segunda contraseña identifica un filtro asignado a la combinación de rango actual, en el que el nuevo filtro se determina en base a un orden de transmisión en el que se señalan los coeficientes de filtro, en el que las contraseñas generadas se usan para señalar a un descodificador la asignación de combinaciones de rango a dichos múltiples filtros.

7. El procedimiento de la reivindicación 6, en el que la segunda métrica se determina en base a la comparación de una medida de

actividad horizontal con una medida de actividad vertical, en el que la actividad cuantifica la actividad asociada al bloque de píxeles.

8. El procedimiento de la reivindicación 7, en el que la medida de actividad horizontal se determina en base a la comparación de un valor de píxel de un píxel actual con un valor de píxel de un píxel vecino izquierdo y un valor de píxel de un píxel vecino derecho, y en el que la medida de la actividad vertical se determina en base a la comparación de un valor de píxel de un píxel actual con un valor de píxel de un píxel vecino superior y un valor de píxel de un píxel vecino inferior.

9. El procedimiento de la reivindicación 7, en el que la primera métrica se determina en base a una suma de la medida de actividad horizontal y de la medida de actividad vertical, en el que la primera métrica cuantifica la actividad asociada al bloque de píxeles.

10. Un aparato que comprende medios para llevar a cabo el procedimiento de una cualquiera de las reivindicaciones 6 a 9.

11. Un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que cuando se ejecutan hacen que uno o más procesadores lleven a cabo el procedimiento de una cualquiera de las reivindicaciones 1 a 4 o 6 a 9.

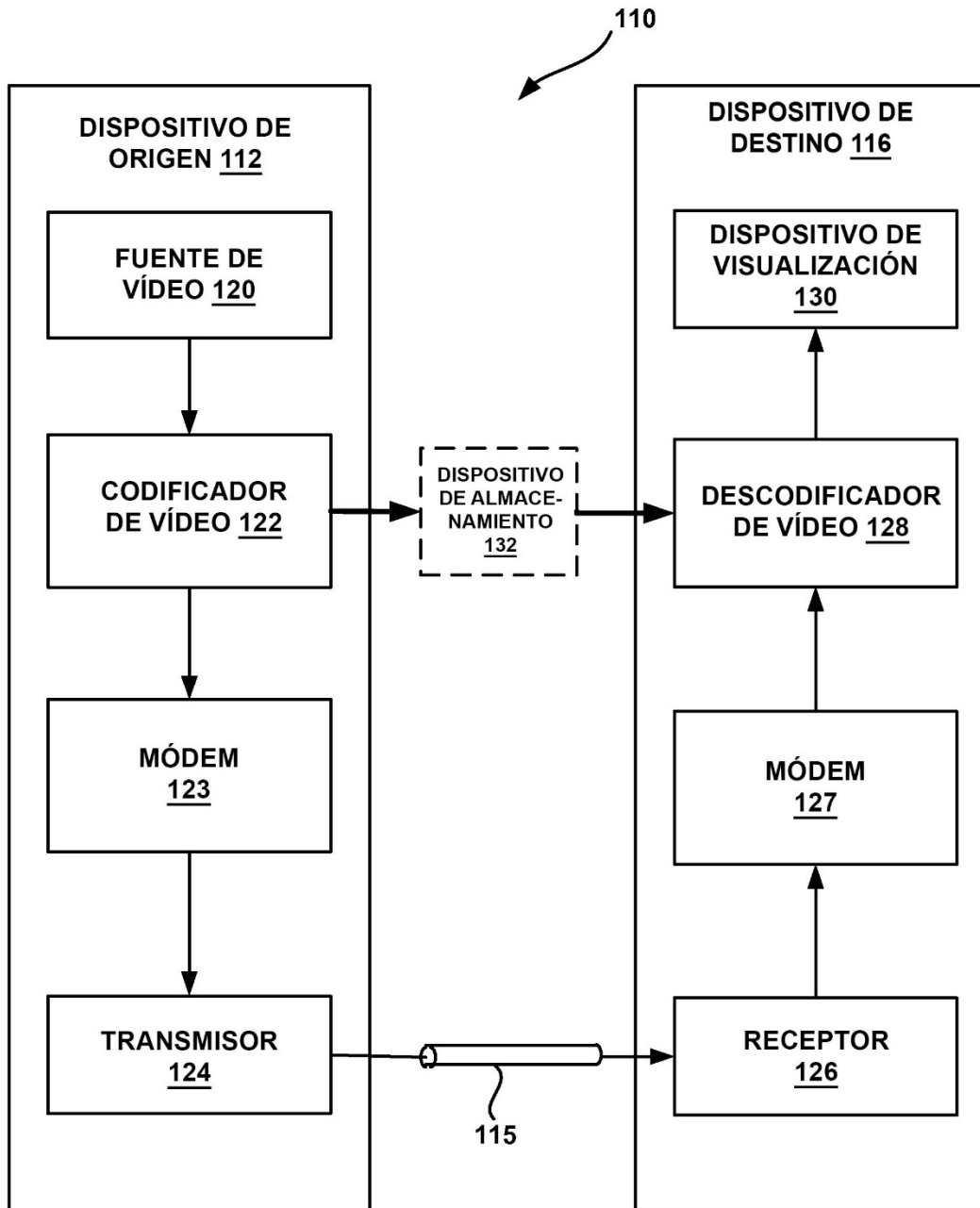


FIG. 1

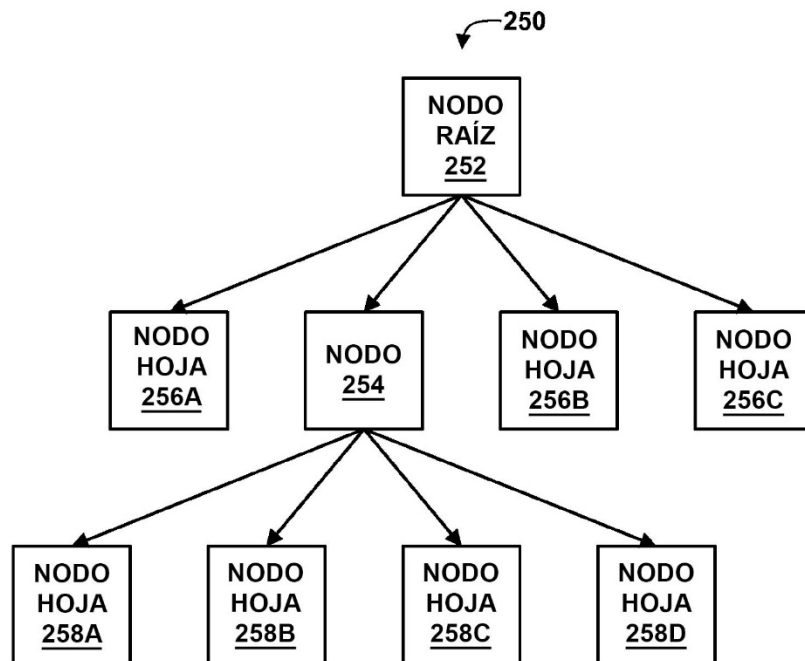


FIG. 2A

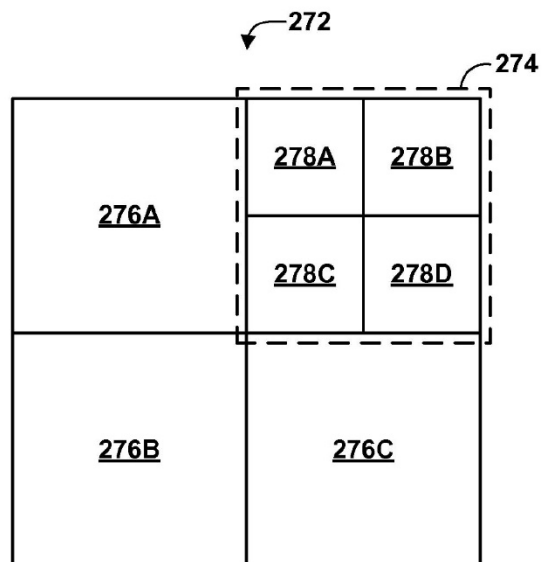


FIG. 2B

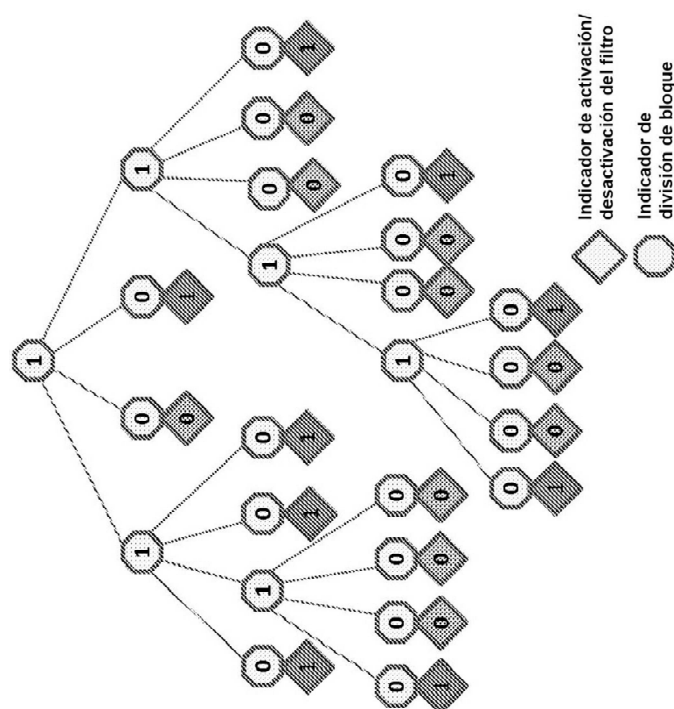


FIG. 2D

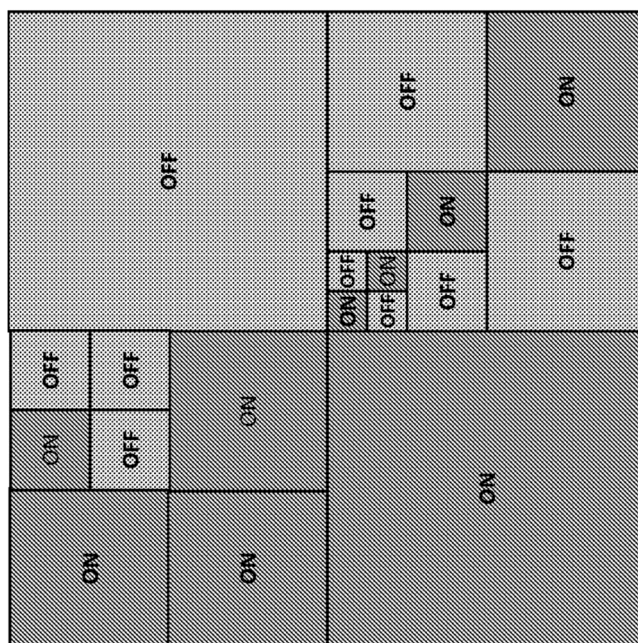
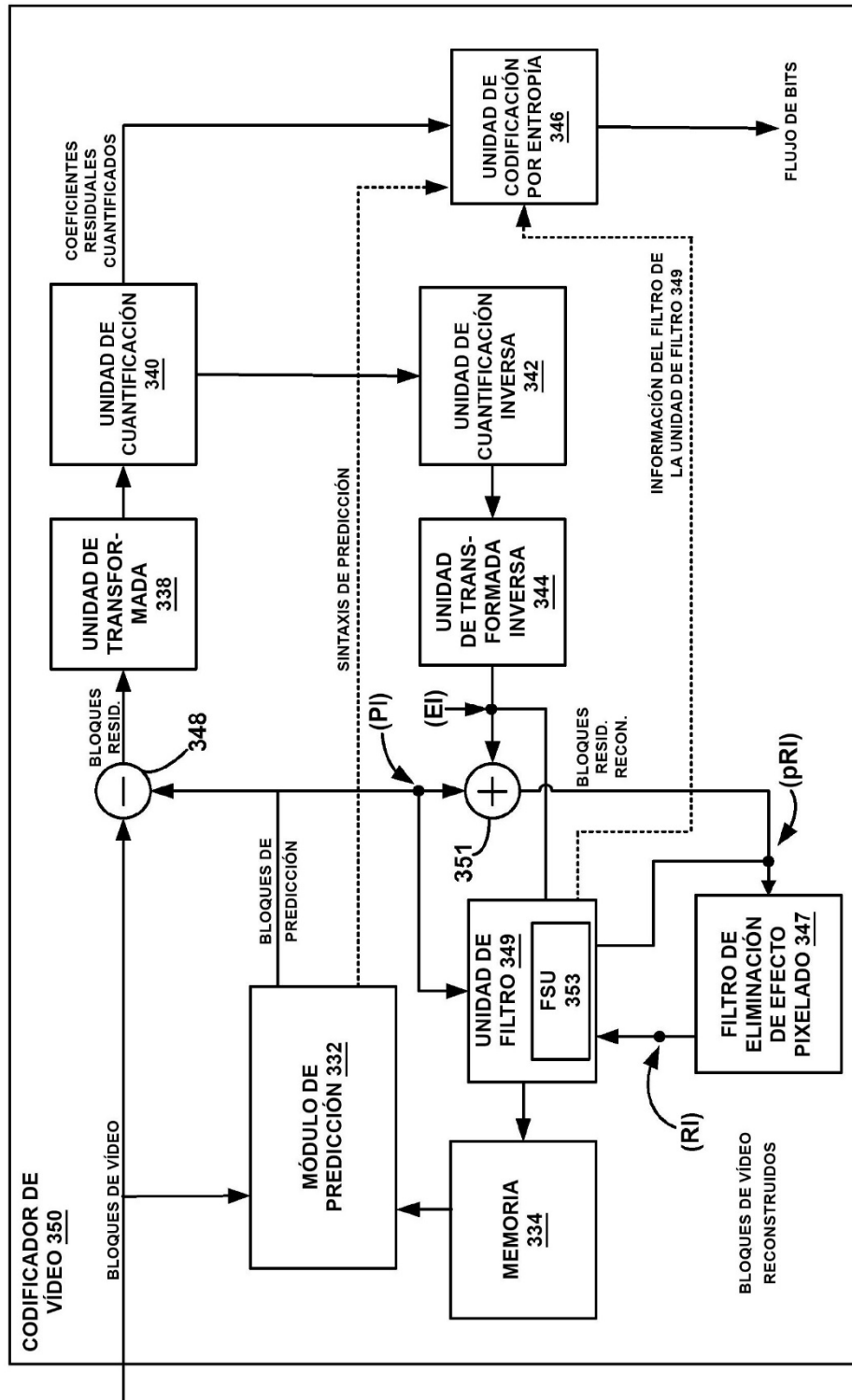


FIG. 2C



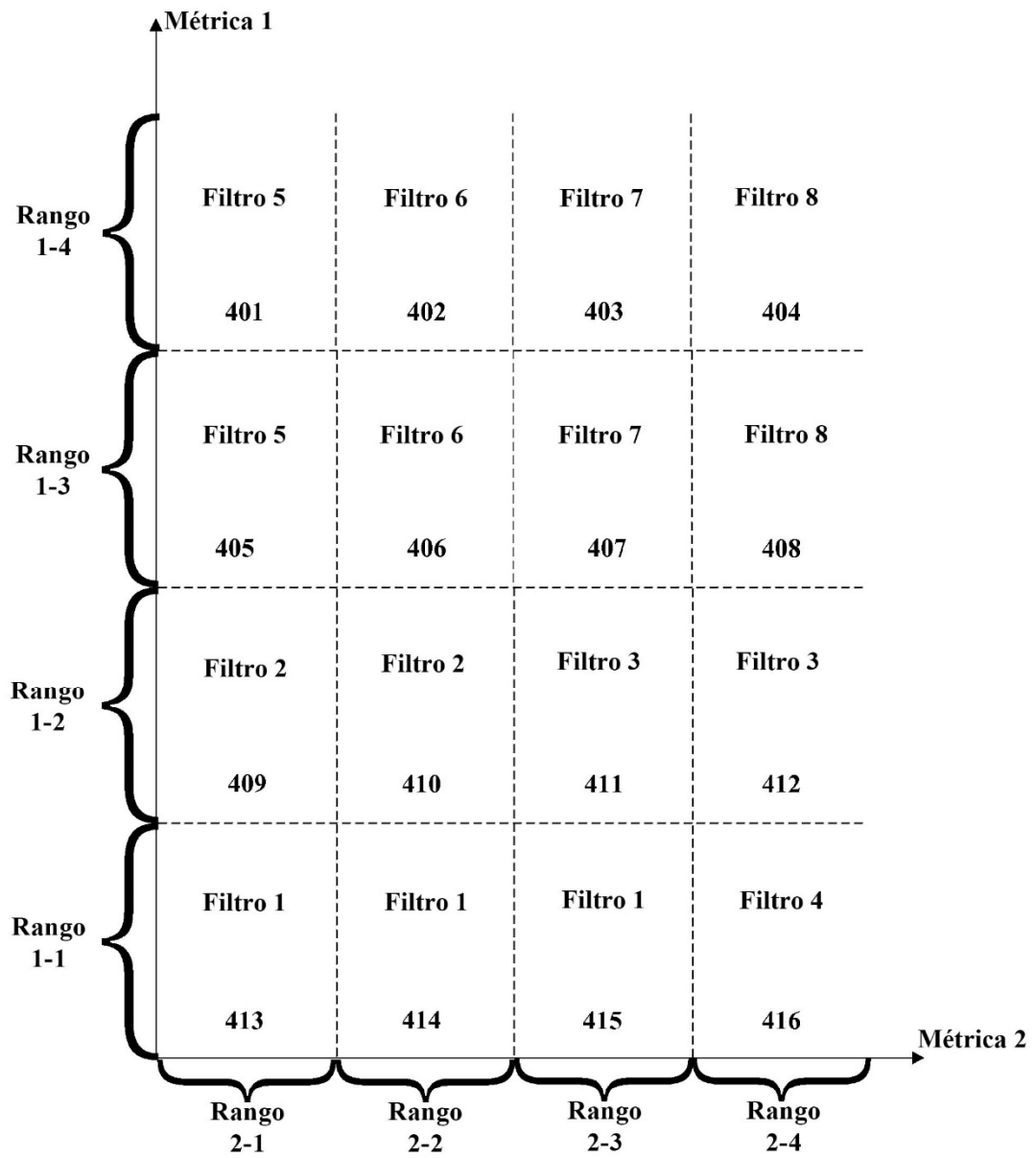


FIG. 4A

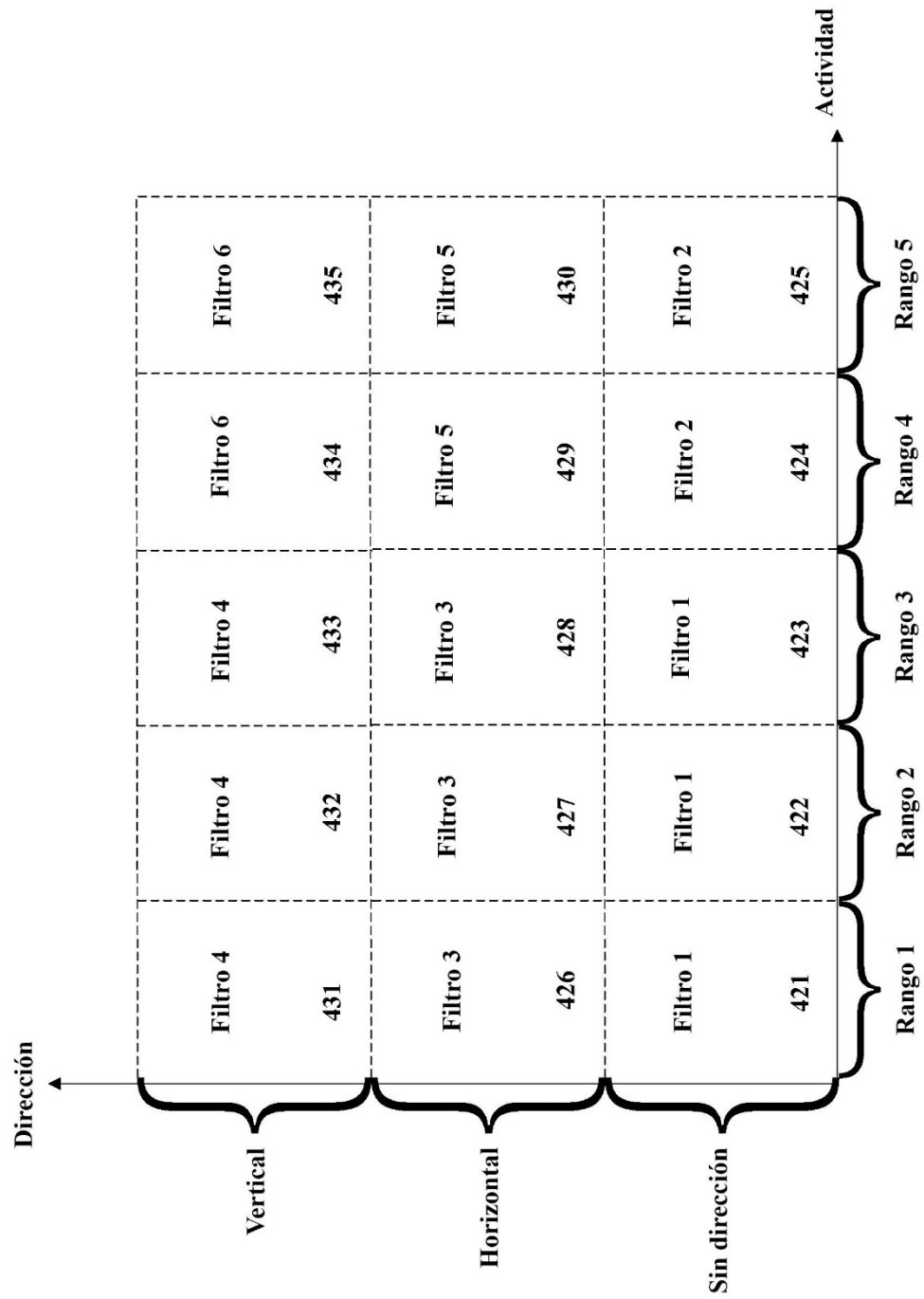


FIG. 4B

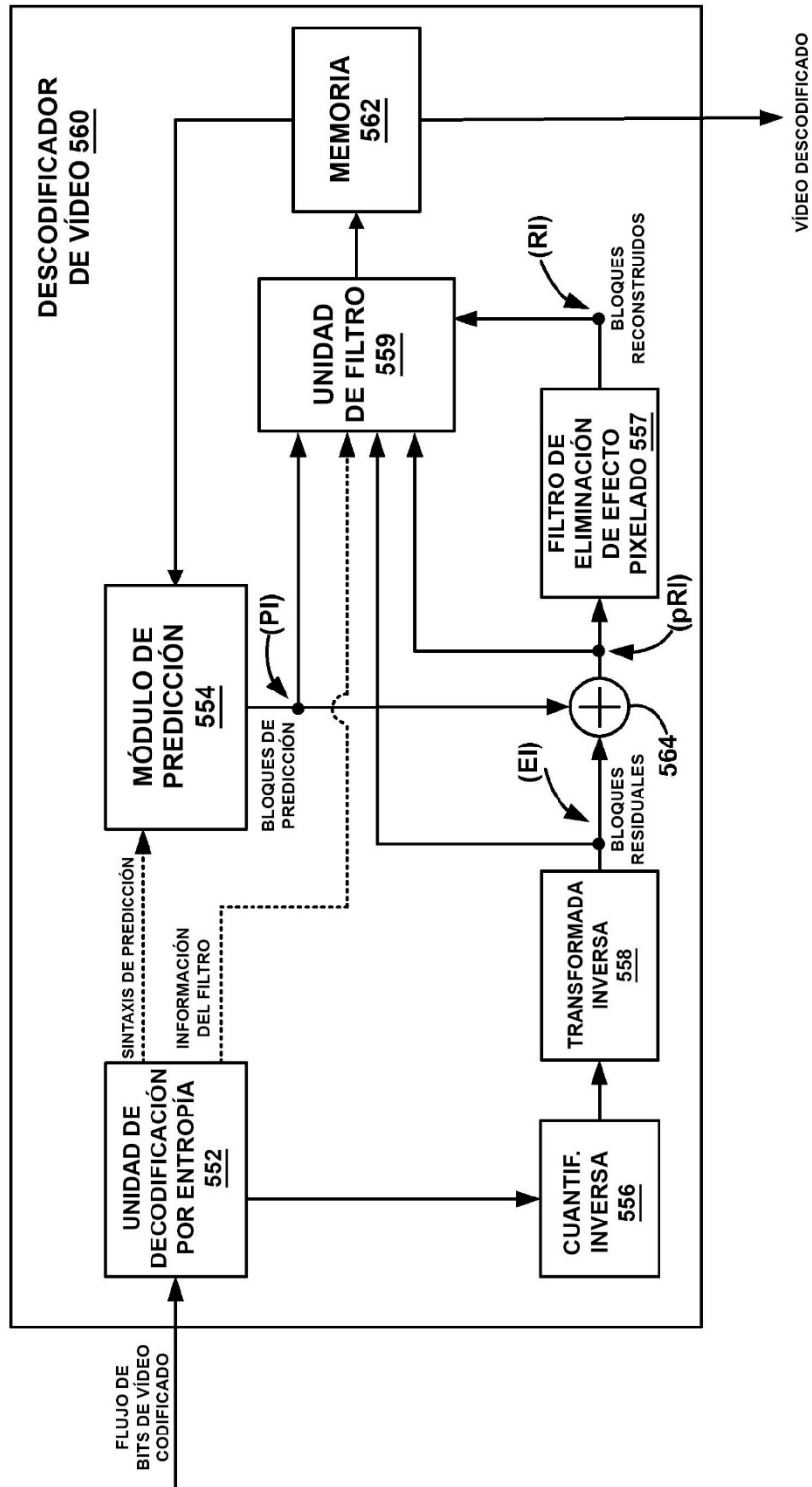


FIG. 5

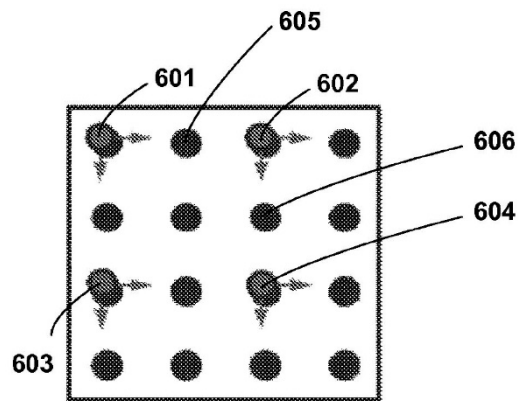


FIG. 6A

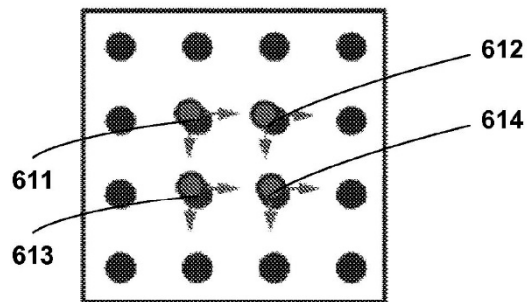


FIG. 6B

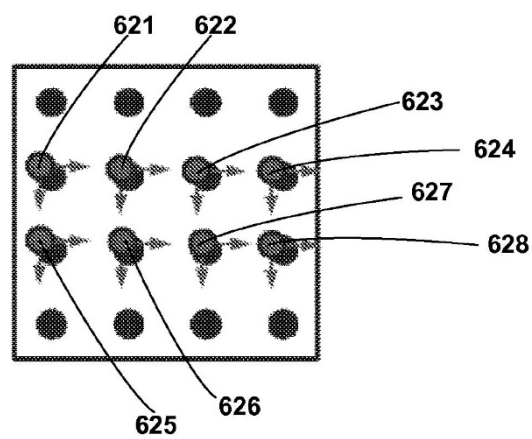


FIG. 6C

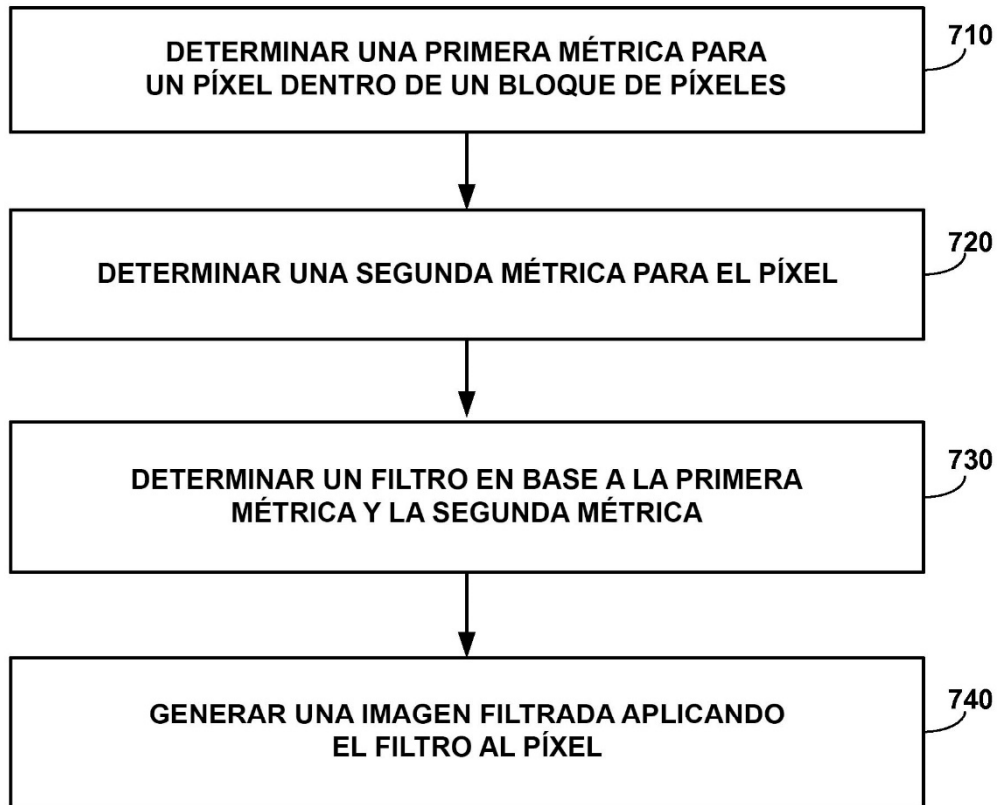


FIG. 7

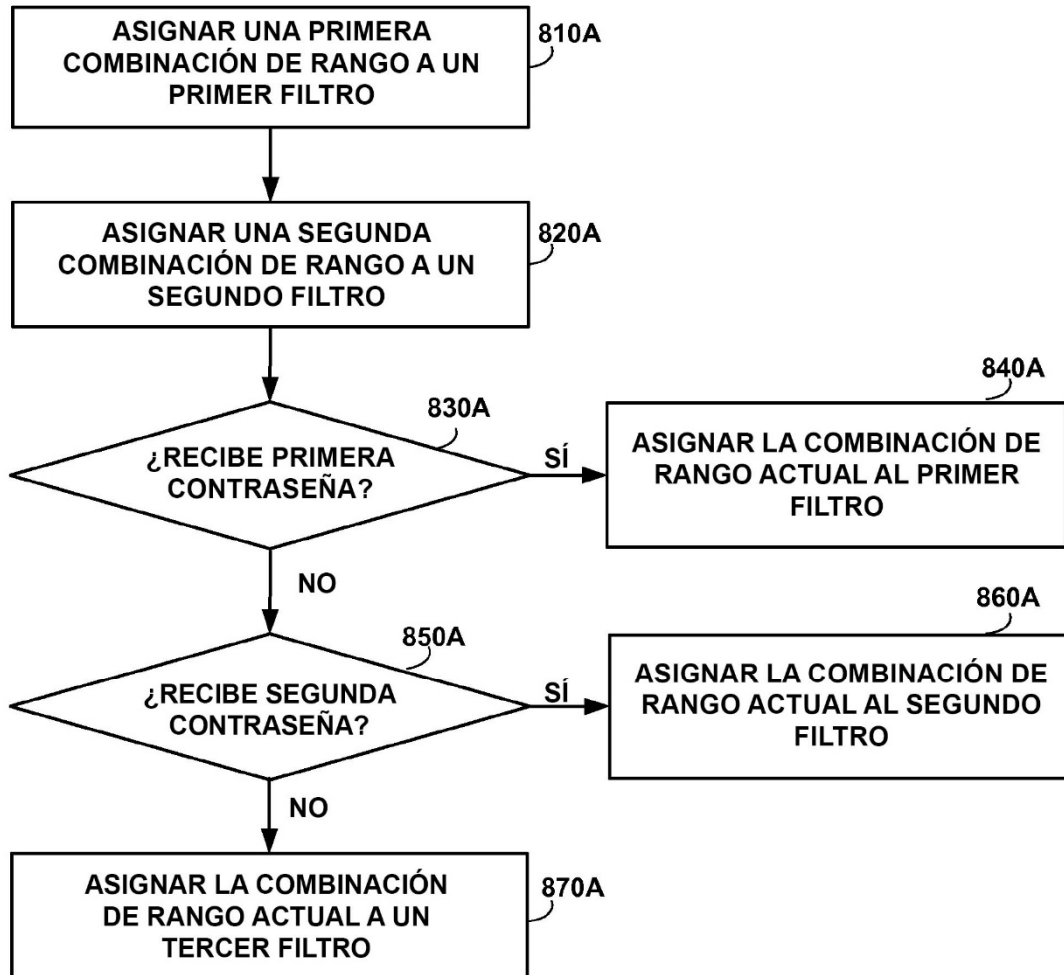


FIG. 8A

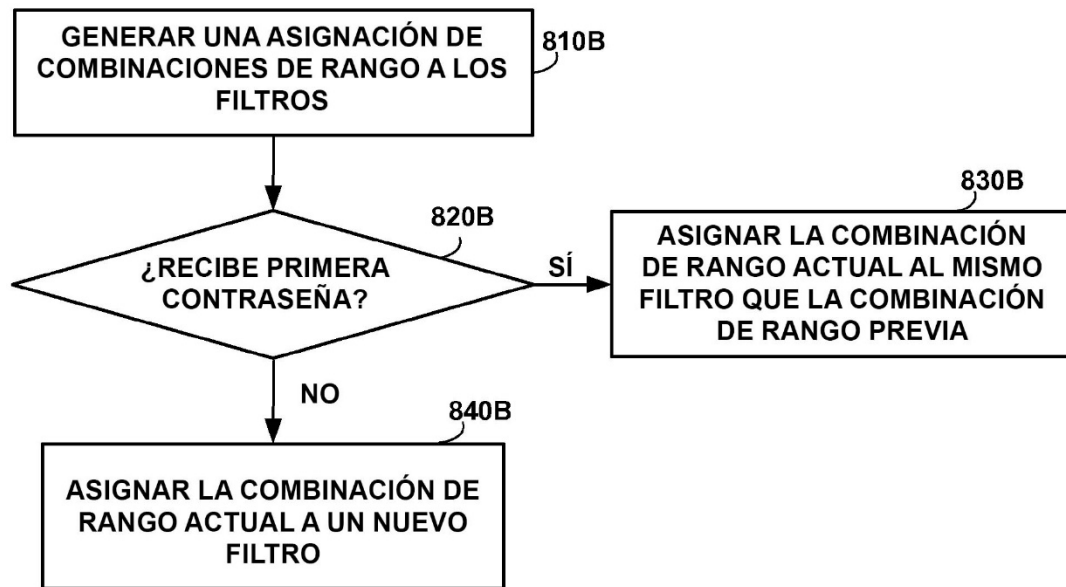


FIG. 8B

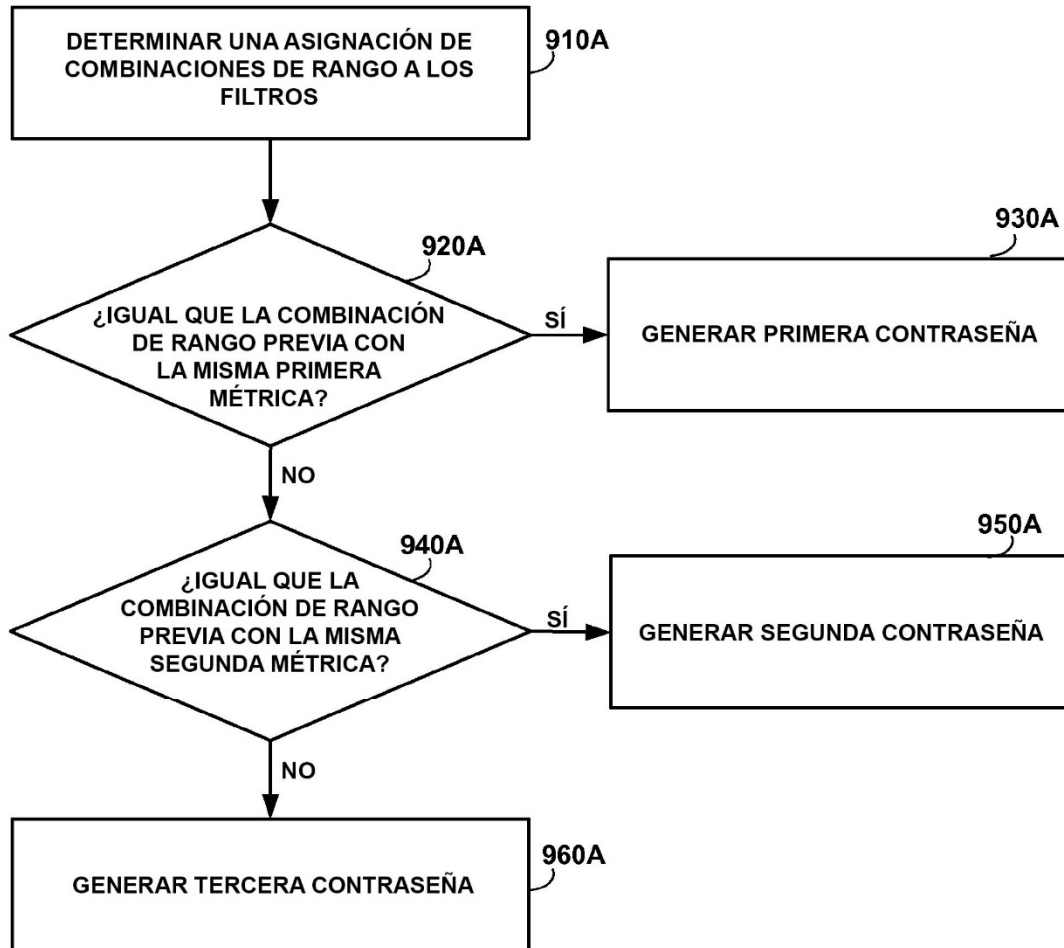


FIG. 9A

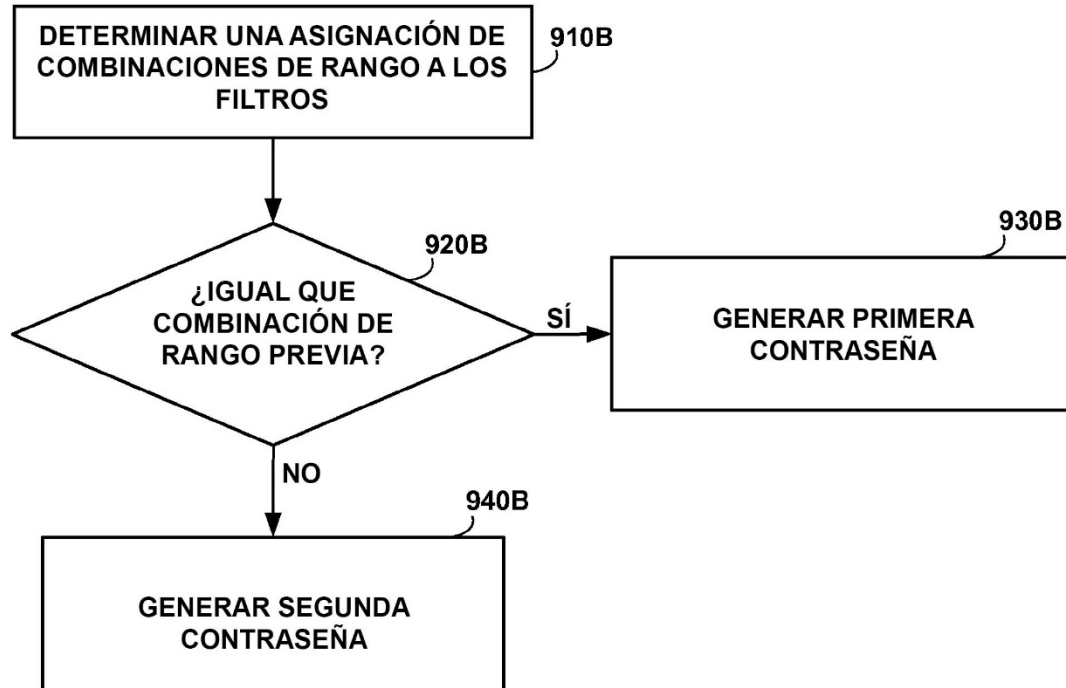


FIG. 9B

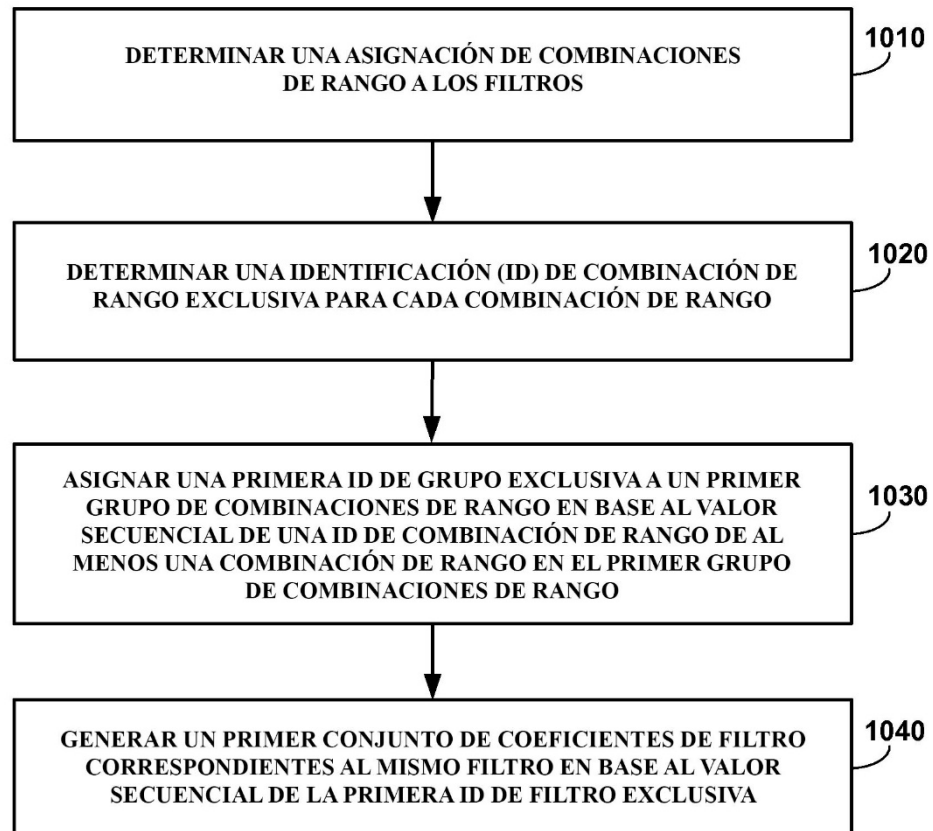


FIG. 10

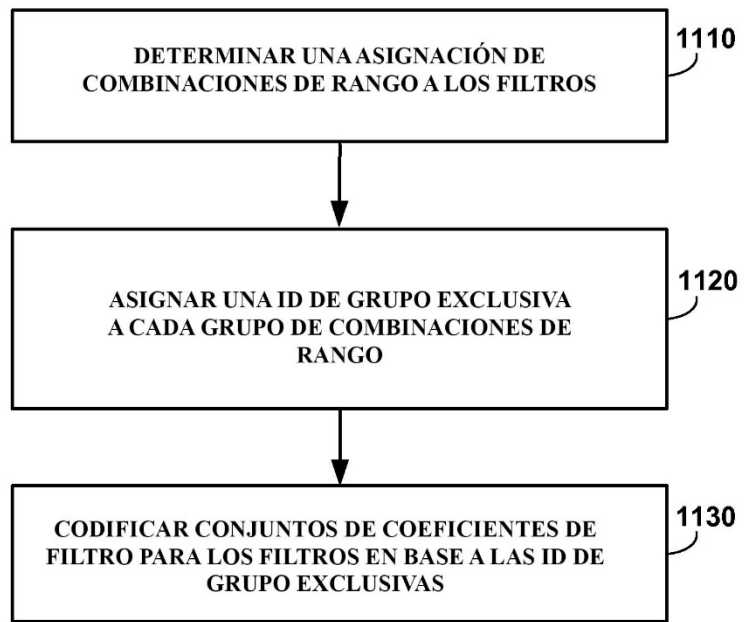


FIG. 11