

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 980 868**

51 Int. Cl.:

H04N 19/117 (2014.01)
H04N 19/169 (2014.01)
H04N 19/52 (2014.01)
H04N 19/70 (2014.01)
H04N 19/82 (2014.01)
G06V 20/40 (2012.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **11.03.2020** **PCT/US2020/022080**
87 Fecha y número de publicación internacional: **17.09.2020** **WO20185883**
96 Fecha de presentación y número de la solicitud europea: **11.03.2020** **E 20769625 (3)**
97 Fecha y número de publicación de la concesión europea: **24.04.2024** **EP 3931748**

54 Título: **Direcciones de segmento basadas en sub-imagen en codificación de vídeo**

30 Prioridad:

11.03.2019 US 201962816751 P
29.03.2019 US 201962826659 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.10.2024

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building Bantian Longgang District
Shenzhen, Guangdong 518129, CN

72 Inventor/es:

WANG, YE-KUI;
CHEN, JIANLE y
HENDRY, FNU

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 980 868 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

Direcciones de segmento basadas en sub-imagen en codificación de vídeo

5 **CAMPO DE LA INVENCION**

La presente divulgación generalmente está relacionada con la codificación de vídeo, y específicamente está relacionada con la codificación de sub-imágenes de imágenes en la codificación de vídeo.

10 **ANTECEDENTES DE LA INVENCION**

La cantidad de datos de vídeo necesarios para mostrar incluso vídeo relativamente corto puede ser sustancial, lo cual puede tener como resultado dificultades cuando los datos van a ser transmitidos por secuencias o de otra manera comunicados a través de una red de comunicaciones con capacidad de ancho de banda limitado. Por lo tanto, los datos de vídeo generalmente son comprimidos antes de ser comunicados a través de las redes de telecomunicaciones de módem. El tamaño de un vídeo también podría ser un problema cuando el vídeo es almacenado en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Dispositivos de compresión de vídeo con frecuencia utilizan software y/o hardware en la fuente para codificar los datos de vídeo antes de la transmisión o almacenamiento, disminuyendo así la cantidad de datos necesarios para representar imágenes de vídeo digitales. Los datos comprimidos son entonces recibidos en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de red limitados y demandas cada vez mayores de calidad de vídeo superior, son deseables técnicas de compresión y descompresión mejoradas que mejoren la relación de compresión con poco a ningún sacrificio en la calidad de las imágenes.

25 El documento ZHOU (TI) M: "AHG4: Enable parallel decoding with tiles", 100. REUNIÓN MPEG; 30-4-2012 - 4-5-2012; GINEBRA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), n° m24357, 24 de abril de 2012 (2012-04-24), XP030052702 divulga la señalización de ID de sub-imagen en el encabezado de segmento para aplicaciones de baja latencia.

30 El documento WO 2019/009473 A1 (LG ELECTRONICS INC [KR]) del 10 de enero de 2019 (2019-01-10) divulga información de índice MCTS que indica el orden de los MCTS que pueden ser señalizada para el ajuste del encabezado de segmento de corte cuando una sub-imagen incluye una pluralidad de MCTS.

SUMARIO

35 La invención se establece en las reivindicaciones adjuntas.

BREVE DESCRIPCIÓN DE LAS FIGURAS

40 Para un entendimiento más completo de esta divulgación, ahora se hace referencia a la siguiente descripción, tomada en conexión con los dibujos acompañantes y la descripción detallada, en donde números de referencia similares representan partes similares.

La figura 1 es un gráfico de flujo de un método ejemplar de codificación de una señal de vídeo.

45 La figura 2 es un diagrama esquemático de un sistema de codificación y decodificación (códec) ejemplar para codificación de vídeo.

La figura 3 es un diagrama esquemático que ilustra un codificador de vídeo ejemplar.

La figura 4 es un diagrama esquemático que ilustra un decodificador de vídeo ejemplar.

50 La figura 5A es un diagrama esquemático que ilustra un ejemplo de imagen dividida en sub-imágenes.

La figura 5B es un diagrama esquemático que ilustra un ejemplo de sub-imagen dividida en segmentos.

La figura 5C es un diagrama esquemático que ilustra un ejemplo de segmento dividido en mosaicos.

La figura 5D es un diagrama esquemático que ilustra un ejemplo de segmento dividido en unidades de árbol de codificación (CTUs).

La figura 6 es un diagrama esquemático que ilustra un ejemplo de inter-predicción unidireccional.

55 La figura 7 es un diagrama esquemático que ilustra un ejemplo de inter-predicción bidireccional.

La figura 8 es un diagrama esquemático que ilustra un ejemplo de codificación de un bloque actual basado en vectores de movimiento candidatos de bloques codificados adyacentes.

La figura 9 es un diagrama esquemático que ilustra un ejemplo de patrón para determinar una lista de candidatos de vectores de movimiento.

60 La figura 10 es un diagrama de bloques que ilustra un ejemplo de filtro en-bucle.

La figura 11 es un diagrama esquemático que ilustra un ejemplo de corriente de bits que contiene parámetros de herramienta de codificación para soportar la decodificación de una sub-imagen de una imagen.

La figura 12 es un diagrama esquemático de un ejemplo de dispositivo de codificación de vídeo.

La figura 13 es un gráfico de flujo de un ejemplo de un método para codificar una secuencia de vídeo en una corriente de bits direccionando un segmento con relación a una sub-imagen.

La figura 14 es un gráfico de flujo de un ejemplo de método de decodificación de una secuencia de vídeo a partir de una corriente de bits cuando un segmento es direccionado con relación a una sub-imagen.

La figura 15 es un diagrama esquemático de un ejemplo de sistema para codificar una secuencia de vídeo de imágenes en una corriente de bits cuando un segmento es direccionado con relación a una sub-imagen.

DESCRIPCIÓN DETALLADA DE LA INVENCION

Se debiera entender al inicio de que aunque a continuación se proporciona una implementación ilustrativa de una o más realizaciones, los sistemas y/o métodos divulgados pueden ser implementados utilizando cualquier número de técnicas, ya sea actualmente conocidas o en existencia. La divulgación de ninguna manera debiera quedar limitada a las implementaciones, dibujos y técnicas ilustradas a continuación, incluyendo los diseños e implementaciones ejemplares ilustrados y descritos aquí, pero pueden ser modificados dentro del alcance de las reivindicaciones anexas.

Los siguientes acrónimos son utilizados aquí, Filtro de Bucle Adaptable (ALF), Bloque de Árbol de Codificación (CTB), Unidad de Árbol de Codificación (CTU), Unidad de Codificación (CU), Secuencia de Vídeo Codificada (CVS), Equipo de Expertos de Vídeo Conjunto (JVET), Conjunto de Mosaicos de Movimiento Restringido (MCTS), Unidad de Transferencia Máxima (MTU), Capa de Abstracción de Red (NAL), Conteo de Orden de Imágenes (POC), Carga Útil de Secuencia de Bytes Sin Procesar (RBSP), Compensación de Muestra Adaptable (SAO), Conjunto de Parámetros de Secuencia (SPS), Predicción de Vector de Movimiento Temporal (TMVP), Codificación de Vídeo Versátil (VVC), y Bosquejo de Trabajo (WD).

Se pueden emplear muchas técnicas de compresión de vídeo para reducir el tamaño de archivos de vídeo con pérdida mínima de datos. Por ejemplo, técnicas de compresión de vídeo pueden incluir la ejecución de predicción espacial (por ejemplo, intra-imagen) y/o predicción temporal (por ejemplo, inter-imagen) para reducir o remover redundancia de datos en secuencias de vídeo. Para codificación de vídeo basada en bloque, un segmento de vídeo (por ejemplo, una imagen de vídeo o una porción de una imagen de vídeo) puede ser dividido en bloques de vídeo, los cuales también pueden ser referidos como bloques de árbol, bloques de árbol de codificación (CTBs), unidades de árbol de codificación (CTUs), unidades de codificación (CUs), y/o nodos de codificación. Los bloques de vídeo en un segmento intra-codificado (I) de una imagen son codificados utilizando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Bloques de vídeo en un segmento inter-codificado de predicción unidireccional (P) o predicción bidireccional (B) de una imagen pueden ser codificados empleando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden ser referidas como cuadros y/o fotografías, e imágenes de referencia pueden ser referidas como cuadros de referencia y/o fotografías de referencia. La predicción espacial o temporal tiene como resultado un bloque predictivo que representa un bloque de imagen. Datos residuales representan diferencias de pixel entre el bloque de imagen original y el bloque predictivo. Por consiguiente, un bloque inter-codificado es codificado de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado es codificado de acuerdo con un modo de intra-codificación y los datos residuales. Para compresión adicional, los datos residuales pueden ser transformados del dominio de pixel a un dominio de transformada. Estos tienen como resultado coeficientes de transformada residuales, los cuales pueden ser cuantificados. Los coeficientes de transformada cuantificados inicialmente pueden ser acomodados en un arreglo bidimensional. Los coeficientes de transformada cuantificados pueden ser escaneados para producir un vector uni-dimensional de coeficientes de transformada. La codificación entrópica puede ser aplicada para lograr incluso más compresión. Dichas técnicas de compresión de vídeo se analizan con mayor detalle a continuación.

Para garantizar que un vídeo codificado pueda ser decodificado con precisión, el vídeo es codificado y decodificado de acuerdo con normas de codificación de vídeo correspondientes. Normas de codificación de vídeo incluyen el Sector de Estandarización de Unión Internacional de Telecomunicaciones (ITU) (ITU-T) H.261, Organización Internacional para el Grupo de Expertos de Imágenes en Movimiento de la Comisión Electrotécnica de Estandarización/Internacional (ISO/IEC) (MPEG)-1 Parte 2, ITU-T H.262 o ISO/IEC MPEG-2 Parte 2, ITU-T H.263, ISO/IEC MPEG-4 Parte 2, Codificación de Vídeo Avanzada (AVC), también conocida como ITU-T H.264 o ISO/IEC MPEG-4 Parte 10, y Codificación de Vídeo de Alta Eficiencia (HEVC), también conocido como ITU-T H.265 o MPEG-H Parte 2. AVC incluye extensiones tales como Codificación de Vídeo Escalable (SVC), Codificación de Vídeo Multivista (MVC) y Codificación de Vídeo Multivista más Profundidad (MVC+D), y AVC (3D-AVC) tridimensional (3D). HEVC incluye extensiones tales como HEVC Escalable (SHVC), HEVC Multivista (MV-HEVC), y 3D HEVC (3D-HEVC). El equipo de expertos de vídeo conjunto (JVET) de ITU-T e ISO/IEC ha comenzado a desarrollar una norma de codificación de vídeo referida como Codificación de Vídeo Versátil (VVC). VVC es incluida en un Bosquejo de

Trabajo (WD), el cual incluye JVET-M1001-v6 que proporciona una descripción de algoritmo, una descripción del lado del codificador del VVC WD, y software de referencia.

A fin de codificar una imagen de vídeo, la imagen primero es dividida, y las divisiones son codificadas en una corriente de bits. Varios esquemas de división de imagen están disponibles. Por ejemplo, una imagen puede ser dividida en segmentos regulares, segmentos dependientes, mosaicos, y/o de acuerdo con Procesamiento Paralelo de Frente de onda (WPP). Por simplicidad, HEVC restringe los codificadores de manera que solamente segmentos regulares, segmentos dependientes, mosaicos, WPP, y combinaciones de los mismos pueden ser utilizados cuando se divide un segmento en grupos de CTBs para codificación de vídeo. Dicha división puede ser aplicada para soportar emparejamiento de tamaño de Unidad de Transferencia Máxima (MTU), procesamiento paralelo, y retraso extremo-a-extremo reducido. MTU denota la cantidad máxima de datos que puede ser transmitida en un solo paquete. Si una carga útil de paquete es en exceso de la MTU, esa carga útil es dividida en dos paquetes a través de un proceso llamado fragmentación.

Un segmento regular, también referido simplemente como un segmento, es una porción dividida de una imagen que puede ser reconstruida independientemente de otros segmentos regulares dentro de la misma imagen, no obstante algunas interdependencias debido a las operaciones de filtración de bucle. Cada segmento regular es encapsulado en su propia unidad de Capa de Abstracción de Red (NAL) para transmisión. Además, la predicción en-imagen (predicción intra muestra, predicción de información de movimiento, predicción de modo de codificación) y la dependencia de codificación entrópica a través de límites de segmento pueden ser inhabilitadas para soportar la reconstrucción independiente. Dicha reconstrucción independiente soporta paralelización. Por ejemplo, paralelización basada en segmento regular emplea comunicación mínima inter-procesador o inter-núcleo. Sin embargo, debido a que cada segmento regular es independiente, cada segmento está asociado con una cabecera de segmento separada. El uso de segmentos regulares puede incurrir en una sobrecarga de codificación sustancial debido al costo en bits de la cabecera de segmento para cada segmento y debido a la falta de predicción a través de los límites de segmento. Además, se pueden emplear segmentos regulares para soportar el emparejamiento para requerimientos de tamaño de MTU. Específicamente, debido a que un segmento regular es encapsulado en una unidad NAL separada y puede ser independientemente codificado, cada segmento regular debiera ser más pequeño que la MTU en esquemas MTU para evitar la ruptura del segmento en múltiples paquetes. Debido a esto, el objetivo de la paralelización y el objetivo del emparejamiento de tamaño de MTU pueden colocar demandas contradictorias para un despliegue de segmento en una imagen.

Segmentos dependientes son similares a segmentos regulares, pero tienen cabeceras de segmento acortadas y permiten la división de los límites de bloque de árbol de imagen sin romper la predicción en-imagen. Por consiguiente, segmentos dependientes permiten que un segmento regular sea fragmentado en múltiples unidades NAL, lo cual proporciona retraso extremo-a-extremo reducido al permitir que una parte de un segmento regular sea enviada antes que se complete la codificación de todo el segmento regular.

Un mosaico es una porción dividida de una imagen creada por límites horizontales y verticales que crean columnas y filas de mosaicos. Los mosaicos pueden ser codificados en un orden de escaneo de trama (derecha a izquierda y superior a inferior). El orden de escaneo de CTBs es local dentro de un mosaico. Por consiguiente, CTBs en un primer mosaico son codificados en el orden de escaneo de trama, antes de avanzar a los CTBs en el siguiente mosaico. Similar a segmentos regulares, los mosaicos rompen las dependencias de predicción en-imagen así como dependencias de decodificación entrópica. Sin embargo, los mosaicos pueden no ser incluidos en unidades NAL individuales, y por lo tanto los mosaicos pueden no ser utilizados para emparejamiento de tamaño de MTU. Cada mosaico puede ser procesado por un procesador/núcleo, y la comunicación inter-procesador/inter-núcleo empleada para predicción en-imagen entre unidades de procesamiento que decodifican mosaicos vecinos puede estar limitada a transmitir una cabecera de segmento compartida (cuando mosaicos adyacentes están en el mismo segmento), y ejecutar filtración de bucle relacionada con compartir muestras reconstruidas y metadatos. Cuando se incluye más de un mosaico en un segmento, la compensación de bytes de punto de entrada para cada mosaico diferente a la primera compensación de punto de entrada en el segmento puede ser señalizada en la cabecera de segmento. Para cada segmento y mosaico, se debiera cumplir al menos una de las siguientes condiciones: 1) todos los bloques de árbol codificados en un segmento pertenecen al mismo mosaico; y 2) todos los bloques de árbol codificados en un mosaico pertenecen al mismo segmento.

En WPP, la imagen es dividida en filas sencillas de CTBs. Mecanismos de decodificación entrópica y predicción pueden utilizar datos de CTBs en otras filas. Se vuelve posible un procesamiento paralelo a través de decodificación paralela de filas de CTB. Por ejemplo, una fila actual puede ser decodificada en paralelo con una fila precedente. Sin embargo, la decodificación de la fila actual es retrasada del proceso de decodificación de las filas precedentes por dos CTBs. Este retraso asegura que los datos relacionados con el CTB de encima y el CTB de encima y a la derecha del CTB actual en la fila actual estén disponibles antes que el CTB actual sea codificado. Este enfoque aparece como un frente de onda cuando es representado gráficamente. Este inicio escalonado permite la paralelización hasta con tantos procesadores/núcleos como la imagen contiene filas de CTB. Debido a que está

permitida la predicción en-imagen entre filas de bloque de árbol vecinas dentro de una imagen, la comunicación inter-procesador/inter-núcleo para permitir la predicción en-imagen puede ser sustancial. La división WPP considera los tamaños de unidad NAL. Por lo tanto, WPP no soporta el emparejamiento de tamaño MTU. Sin embargo, segmentos regulares pueden ser utilizados en conjunto con WPP, con cierta sobrecarga de codificación, para implementar el emparejamiento de tamaño de MTU según se desee.

Los mosaicos también pueden incluir conjuntos de mosaicos de movimiento restringido. Un conjunto de mosaicos de movimiento restringido (MCTS) es un conjunto de mosaicos diseñado de manera que vectores de movimiento asociados están restringidos a apuntar a ubicaciones de muestra completa dentro del MCTS y a ubicaciones de muestra fraccional que solamente requieren ubicaciones de muestra completa dentro del MCTS para interpolación. Además, no está autorizado el uso de candidatos de vector de movimiento para predicción de vector de movimiento temporal derivada de bloques fuera del MCTS. De esta manera, cada MCTS puede ser decodificado independientemente sin la existencia de mosaicos no incluidos en el MCTS. Mensajes de información de mejoras complementarias (SEI) de MCTSs temporales pueden ser utilizados para indicar la existencia de MCTSs en la corriente de bits y señalar los MCTSs. El mensaje SEI de MCTSs proporciona información complementaria que puede ser utilizada en la extracción de sub-corriente de bits MCTS (especificada como parte de la semántica del mensaje SEI) para generar una corriente de bits de conformación para un conjunto MCTS. La información incluye un número de conjuntos de información de extracción, cada uno definiendo un número de conjuntos MCTS y conteniendo bytes de carga útil de secuencia de bytes sin procesar (RBSP) del conjunto de parámetros de vídeo de reemplazo (VPSs), conjuntos de parámetros de secuencia (SPSs), y conjuntos de parámetros de imagen (PPSs) que van a ser utilizados durante el proceso de extracción de sub-corriente de bits MCTS. Cuando se extrae una sub-corriente de bits de acuerdo con el proceso de extracción de sub-corriente de bits MCTS, conjuntos de parámetros (VPSs, SPSs, y PPSs) pueden ser reescritos o reemplazados, y cabeceras de segmento pueden ser actualizadas debido a que uno o todos los elementos de sintaxis relacionados con dirección de segmento (incluyendo `primer_porción_segmento_en_imagen_indicador` y `porción_slice_address`) pueden emplear diferentes valores en la sub-corriente de bits extraída.

Las imágenes también pueden ser divididas en una o más sub-imágenes. La división de una imagen en una sub-imagen puede permitir que diferentes porciones de una imagen sean tratadas de manera diferente desde un punto de vista de la codificación. Por ejemplo, una sub-imagen puede ser extraída y desplegada sin extraer las otras sub-imágenes. Como otro ejemplo, diferentes sub-imágenes pueden ser desplegadas a diferentes resoluciones, reposicionadas con relación entre sí (por ejemplo, en aplicaciones de teleconferencia), o de otra manera codificadas como imágenes separadas aun cuando las sub-imágenes colectivamente contienen datos de una imagen común.

Una implementación ejemplar de sub-imágenes es de la siguiente manera. Una imagen puede ser dividida en una o más sub-imágenes. Una sub-imagen es un conjunto rectangular o cuadrado de grupos de segmentos/mosaicos que comienzan con un grupo de segmentos/mosaicos que tiene una dirección igual a cero. Cada sub-imagen se puede referir a un PPS diferente, y por lo tanto, cada sub-imagen puede emplear un mecanismo de división diferente. Las sub-imágenes pueden ser tratadas como imágenes en el proceso de decodificación. Una imagen de referencia actual utilizada para decodificar una sub-imagen actual puede ser generada extrayendo un área que recurre a la sub-imagen actual de las imágenes de referencia en la memoria tampón de imágenes decodificadas. El área extraída puede ser una sub-imagen decodificada, y por lo tanto la inter-predicción puede ocurrir entre sub-imágenes del mismo tamaño y la misma ubicación dentro de la imagen. Un grupo de mosaicos puede ser una secuencia de mosaicos en escaneo de trama de mosaico de una sub-imagen. Lo siguiente puede ser derivado para determinar la ubicación de una sub-imagen en una imagen. Cada sub-imagen puede ser incluida en la siguiente ubicación sin ocupar en el orden de escaneo de trama de CTU dentro de una imagen que es lo suficientemente grande para ajustarse a la sub-imagen dentro de los límites de imagen.

Los esquemas de sub-imagen empleados por varios sistemas de codificación de vídeo incluyen varios problemas que reducen la eficiencia y/o funcionalidad de la codificación. La presente divulgación incluye varias soluciones a dichos problemas. En un primer problema ejemplar, se puede ejecutar inter-predicción de acuerdo con uno de varios modos de inter-predicción. Algunos modos de inter-predicción generan listas de candidatos de predictores de vector de movimiento en ambos el codificador y el decodificador. Esto permite que el codificador señale un vector de movimiento señalizando el índice de la lista de candidatos en lugar de señalar el vector de movimiento completo. Además, algunos sistemas codifican sub-imágenes para extracción independiente. Esto permite que una sub-imagen actual sea decodificada y desplegada sin decodificar información de otras sub-imágenes. Esto puede provocar errores cuando se emplea un vector de movimiento que apunta fuera de la sub-imagen debido a que los datos apuntados por el vector de movimiento no pueden ser decodificados y, por lo tanto, no pueden estar disponibles.

Por consiguiente, en un primer ejemplo, aquí se divulga un indicador que indica que una sub-imagen debiera ser tratada como una imagen. Este indicador es establecido para soportar extracción separada de la sub-imagen. Cuando se establece el indicador, los predictores de vector de movimiento obtenidos de un bloque colocado incluyen

solamente vectores de movimiento que apuntan al interior de la sub-imagen. Cualesquiera predictores de vector de movimiento que apuntan fuera de la sub-imagen son excluidos. Esto garantiza que vectores de movimiento que apuntan fuera de la sub-imagen no sean seleccionados y se evitan errores asociados. Un bloque colocado es un bloque de una imagen diferente de la imagen actual. Predictores de vector de movimiento de bloques en la imagen actual (bloques no-colocados) pueden apuntar fuera de la sub-imagen debido a que otros procesos, tal como filtros de interpolación, pueden evitar errores para dichos predictores de vector de movimiento. Por consiguiente, el presente ejemplo proporciona funcionalidad adicional a un codificador/decodificador de vídeo (códec) evitando errores cuando se ejecuta extracción de sub-imagen.

En un segundo ejemplo, aquí se divulga un indicador que indica que una sub-imagen debiera ser tratada como una imagen. Cuando una sub-imagen actual es tratada como una imagen, la sub-imagen actual debiera ser extraída sin referencia a otras sub-imágenes. Específicamente, el presente ejemplo emplea una función de recorte que es aplicada cuando se aplican filtros de interpolación. Esta función de recorte garantiza que el filtro de interpolación no se base en datos de sub-imágenes adyacentes para mantener la separación entre las sub-imágenes para soportar la extracción separada. Debido a esto, la función de recorte es aplicada cuando el indicador es establecido y un vector de movimiento apunta fuera de la sub-imagen actual. El filtro de interpolación es entonces aplicado a los resultados de la función de recorte. Por consiguiente, el presente ejemplo proporciona funcionalidad adicional a un códec de vídeo evitando errores cuando se ejecuta extracción de sub-imagen. Debido a esto, el primer ejemplo y el segundo ejemplo abordan el primer problema ejemplar.

En un segundo problema ejemplar, sistemas de codificación de vídeo dividen imágenes en sub-imágenes, segmentos, mosaicos, y/o unidades de árbol de codificación, las cuales entonces son divididas en bloques. Dichos bloques son entonces codificados para transmisión hacia un decodificador. La decodificación de dichos bloques puede tener como resultado una imagen decodificada que contiene varios tipos de ruido. Para corregir dichas situaciones, sistemas de codificación de vídeo pueden aplicar varios filtros a través de límites de bloque. Estos filtros pueden remover el bloqueo, el ruido de cuantificación, y otros artefactos de codificación indeseables. Como se observó antes, algunos sistemas codifican sub-imágenes para extracción independiente. Esto permite que una sub-imagen actual sea decodificada y desplegada sin decodificar información de otras sub-imágenes. En dichos sistemas, la sub-imagen puede ser dividida en bloques para codificación. Como tal, límites de bloque a lo largo del borde de sub-imagen se pueden alinear con límites de sub-imagen. En algunos casos, los límites de bloque también se pueden alinear con los límites de mosaico. Se pueden aplicar filtros a través de dichos límites de bloque, y por lo tanto se pueden aplicar a través de límites de sub-imagen y/o límites de mosaico. Esto puede provocar errores cuando una sub-imagen actual es independientemente extraída debido a que el proceso de filtración puede operar en una manera inesperada cuando los datos de una sub-imagen adyacente no están disponibles.

En un tercer ejemplo, aquí se divulga un indicador que controla la filtración al nivel de sub-imagen. Cuando se establece el indicador para una sub-imagen, se pueden aplicar filtros a través del límite de sub-imagen. Cuando no se establece el indicador, los filtros no son aplicados a través del límite de sub-imagen. De esta manera, los filtros pueden ser apagados para sub-imágenes que son codificadas para extracción separada o encendidos para sub-imágenes que son codificadas para despliegue como un grupo. Debido a esto, el presente ejemplo proporciona funcionalidad adicional a un códec de vídeo evitando errores relacionados con filtro cuando se ejecuta extracción de sub-imagen.

En un cuarto ejemplo, aquí se divulga un indicador que puede ser establecido para controlar la filtración al nivel de mosaico. Cuando se establece el indicador para un mosaico, se pueden aplicar filtros a través del límite de mosaico. Cuando no se establece el indicador, los filtros no son aplicados a través del límite de mosaico. De esta manera, Los filtros pueden ser apagados o encendidos para uso en límites de mosaico (por ejemplo, mientras se continúan filtrando las porciones internas del mosaico). Por consiguiente, el presente ejemplo proporciona funcionalidad adicional a un códec de vídeo soportando filtración selectiva a través de límites de mosaico. Debido a esto, el tercer ejemplo y el cuarto ejemplo abordan el segundo problema ejemplar.

En un tercer problema ejemplar, sistemas de codificación de vídeo pueden dividir una imagen en sub-imágenes. Esto permite que diferentes sub-imágenes sean tratadas de manera diferente cuando se codifique el vídeo. Por ejemplo, sub-imágenes pueden ser separadamente extraídas y desplegadas, redimensionadas independientemente con base en los cambios de nivel de aplicación, etc. En algunos casos, se pueden crear sub-imágenes dividiendo una imagen en mosaicos y asignando los mosaicos a las sub-imágenes. Algunos sistemas de codificación de vídeo describen los límites de sub-imagen en términos de mosaicos incluidos en la sub-imagen. Sin embargo, esquemas de formación de mosaicos pueden no ser empleados en algunas imágenes. Por consiguiente, dichas descripciones de límite pueden limitar el uso de sub-imágenes a imágenes que emplean mosaicos.

En un quinto ejemplo, aquí se divulga un mecanismo para señalar límites de sub-imagen en términos de CTBs y/o CTUs. Específicamente, la anchura y altura de una sub-imagen pueden ser señalizadas en unidades de CTBs. También, la posición de la CTU superior izquierda de la sub-imagen puede ser señalizada como una compensación

de la CTU superior izquierda de la imagen tal como se mide en CTBs. Tamaños de CTU y CTB pueden ser establecidos a un valor predeterminado. Por consiguiente, la señalización de las dimensiones y posición de la sub-imagen en términos de CTBs y CTUs proporciona suficiente información para que un decodificador coloque la sub-imagen para despliegue. Esto permite que sub-imágenes sean empleadas incluso cuando los mosaicos no son empleados. También, este mecanismo de señalización evita la complejidad y puede ser codificado utilizando relativamente pocos bits. Debido a esto, el presente ejemplo proporciona funcionalidad adicional a un códec de vídeo al permitir que sub-imágenes sean empleadas independientemente de los mosaicos. Además, el presente ejemplo incrementa la eficiencia de la codificación, y por lo tanto reduce el uso de recursos de procesador, memoria y/o red en el codificador y/o decodificador. Debido a esto, el quinto ejemplo aborda el tercer problema ejemplar.

En un cuarto problema ejemplar, una imagen puede ser dividida en una pluralidad de segmentos para codificación. En algunos sistemas de codificación de vídeo, los segmentos son abordados con base en su posición relativa a la imagen. Aún otros sistemas de codificación de vídeo emplean el concepto de sub-imágenes. Como se observó antes, una sub-imagen puede ser tratada de manera diferente de otras sub-imágenes desde una perspectiva de la codificación. Por ejemplo, una sub-imagen puede ser extraída y desplegada independientemente de otras sub-imágenes. En dicho caso, las direcciones de segmento que son generadas con base en posición de imagen pueden dejar de operar adecuadamente debido a que se omite un número importante de las direcciones de segmento esperadas. Algunos sistemas de codificación de vídeo abordan este problema reescribiendo dinámicamente cabeceras de segmento al momento de la solicitud de cambiar direcciones de segmento para soportar la extracción de sub-imagen. Dicho proceso puede ser intenso en recursos, debido a que este proceso puede ocurrir cada vez que un usuario solicita ver la sub-imagen.

En un sexto ejemplo, aquí se divulgan segmentos que son abordados con relación a la sub-imagen que contiene el segmento. Por ejemplo, la cabecera de segmento puede incluir un identificador de sub-imagen (ID) y una dirección de cada segmento incluido en la sub-imagen. Además, un conjunto de parámetros de secuencia (SPS) puede contener dimensiones de la sub-imagen que pueden ser referenciadas por el ID de sub-imagen. Por consiguiente, la cabecera de segmento no necesita ser reescrita cuando se solicita extracción separada de la sub-imagen. La cabecera de segmento y el SPS contienen suficiente información para soportar la colocación de los segmentos en la sub-imagen para despliegue. Debido a esto, el presente ejemplo incrementa la eficiencia de la codificación y/o evita la reescritura redundante de la cabecera de segmento, y por lo tanto reduce el uso de recursos de procesador, memoria y/o red en el codificador y/o decodificador. Por consiguiente, el sexto ejemplo aborda el cuarto problema ejemplar.

La figura 1 es un gráfico de flujo de un método operativo ejemplar 100 de codificación de una señal de vídeo. Específicamente, una señal de vídeo es codificada en un codificador. El proceso de codificación comprime la señal de vídeo empleando varios mecanismos para reducir el tamaño del archivo de vídeo. Un tamaño de archivo más pequeño permite que el archivo de vídeo comprimido sea transmitido a un usuario, mientras se reduce la sobrecarga de ancho de banda asociada. El decodificador entonces decodifica el archivo de vídeo comprimido para reconstruir la señal de vídeo original para despliegue a un usuario final. El proceso de decodificación generalmente refleja el proceso de codificación para permitir que el decodificador reconstruya de manera consistente la señal de vídeo.

En el paso 101, la señal de vídeo es introducida en el codificador. Por ejemplo, la señal de vídeo puede ser un archivo de vídeo no comprimido almacenado en memoria. Como otro ejemplo, el archivo de vídeo puede ser capturado por un dispositivo de captura de vídeo, tal como una cámara de vídeo, y puede ser codificado para soportar la transmisión en vivo del vídeo. El archivo de vídeo puede incluir tanto un componente de audio como un componente de vídeo. El componente de vídeo contiene una serie de cuadros de imagen que, cuando son vistos en una secuencia, proporciona la impresión visual de movimiento. Los cuadros contienen píxeles que son expresados en términos de luz, referidos aquí como componentes de luminancia (o muestras de luminancia), y color, el cual es referido como componentes de crominancia (o muestras de color). En algunos ejemplos, los cuadros también pueden contener valores de profundidad para soportar visualización tridimensional.

En el paso 103, el vídeo es dividido en bloques. La división incluye subdividir los píxeles en cada cuadro en bloques cuadrados y/o rectangulares para compresión. Por ejemplo, en Codificación de Vídeo de Alta Eficiencia (HEVC) (también conocida como H.265 y MPEG-H Parte 2) el cuadro primero puede ser dividido en unidades de árbol de codificación (CTUs), los cuales son bloques de un tamaño predefinido (por ejemplo, sesenta y cuatro píxeles por sesenta y cuatro píxeles). Las CTUs contienen tanto muestras de luminancia como de crominancia. Los árboles de codificación pueden ser empleados para dividir las CTUs en bloques y después subdividir recursivamente los bloques hasta que se logran las configuraciones que soportan codificación adicional. Por ejemplo, componentes de luminancia de un cuadro pueden ser subdivididos hasta que los bloques individuales contienen valores de iluminación relativamente homogéneos. Además, componentes de crominancia de un cuadro pueden ser subdivididos hasta que los bloques individuales contienen valores de color relativamente homogéneos. Por consiguiente, los mecanismos de división varían dependiendo del contenido de los cuadros de vídeo.

En el paso 105, se emplean varios mecanismos de compresión para comprimir los bloques de imagen divididos en el paso 103. Por ejemplo, se puede emplear inter-predicción y/o intra-predicción. La Inter-predicción está diseñada para sacar ventaja del hecho de que los objetos en una escena común tienden a aparecer en cuadros sucesivos. Por consiguiente, un bloque que muestra un objeto en un cuadro de referencia no necesita describirse repetidamente en cuadros adyacentes. Específicamente, un objeto, tal como una tabla, puede permanecer en una posición constante sobre múltiples cuadros. Por lo tanto, la tabla se describe una vez y los cuadros adyacentes pueden hacer referencia nuevamente al cuadro de referencia. Se pueden emplear mecanismos de emparejamiento de patrón para emparejar objetos sobre múltiples cuadros. Además, objetos en movimiento pueden ser representados a través de múltiples cuadros, por ejemplo debido al movimiento de los objetos o al movimiento de la cámara. Como un ejemplo particular, un vídeo puede mostrar un automóvil que se mueve a través de la pantalla sobre múltiples cuadros. Vectores de movimiento pueden ser empleados para describir dicho movimiento. Un vector de movimiento es un vector bidimensional que proporciona una compensación de las coordenadas de un objeto en un cuadro a las coordenadas del objeto en un cuadro de referencia. Debido a esto, la inter-predicción puede codificar un bloque de imagen en un cuadro actual como un conjunto de vectores de movimiento indicando una compensación de un bloque correspondiente en un cuadro de referencia.

La intra-predicción codifica bloques en un cuadro común. La intra-predicción saca ventaja del hecho de que componentes de luminancia y crominancia tienden a agruparse en un cuadro. Por ejemplo, un parche de verde en una porción de un árbol tiende a quedar colocado junto a parches similares de verde. La intra-predicción emplea múltiples modos de predicción direccional (por ejemplo, treinta y tres en HEVC), un modo plano, y un modo de corriente directa (DC). Los modos direccionales indican que un bloque actual es similar/el mismo que las muestras de un bloque vecino en una dirección correspondiente. El modo plano indica que una serie de bloques a lo largo de una fila/columna (por ejemplo, un plano) pueden ser interpolados con base en bloques vecinos en los bordes de la fila. El modo plano, en efecto, indica una transición suave de luz/color a través de una fila/columna empleando una inclinación relativamente constante en valores cambiantes. El modo DC es empleado para suavización de límite e indica que un bloque es similar/el mismo que un valor promedio asociado con muestras de todos los bloques vecinos asociados con las direcciones angulares de los modos de predicción direccionales. Por consiguiente, bloques de intra-predicción pueden representar bloques de imagen como varios valores de modo de predicción de relación en lugar de los valores reales. Además, bloques de inter-predicción pueden representar bloques de imagen como valores de vector de movimiento en lugar de los valores reales. En cualquier caso, los bloques de predicción pueden no representar exactamente los bloques de imagen en algunos casos. Cualesquiera diferencias son almacenadas en bloques residuales. Las transformadas pueden ser aplicadas a los bloques residuales para comprimir adicionalmente el archivo.

En el paso 107, se pueden aplicar varias técnicas de filtración. En HEVC, los filtros son aplicados de acuerdo con un esquema de filtración en-bucle. La predicción basada en bloque antes analizada puede tener como resultado la creación de imágenes de bloqueo en el decodificador. Además, el esquema de predicción basada en bloque puede codificar un bloque y después puede reconstruir el bloque codificado para posterior uso como un bloque de referencia. El esquema de filtración en-bucle iterativamente aplica filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptable, y filtros de compensación adaptativa de muestra (SAO) a los bloques/cuadros. Estos filtros mitigan dichos artefactos de bloqueo de manera que el archivo codificado puede ser reconstruido con precisión. Además, estos filtros mitigan los artefactos en los bloques de referencia reconstruidos de manera que los artefactos tienen menos probabilidades de crear artefactos adicionales en bloques subsiguientes que son codificados con base en los bloques de referencia reconstruidos.

Una vez que la señal de vídeo ha sido dividida, comprimida y filtrada, los datos resultantes son codificados en una corriente de bits en el paso 109. La corriente de bits incluye los datos antes analizados así como cualesquiera datos de señalización deseados para soportar la correcta reconstrucción de señal de vídeo en el decodificador. Por ejemplo, dichos datos pueden incluir datos de división, datos de predicción, bloques residuales, y varios indicadores que proporcionan instrucciones de codificación al decodificador. La corriente de bits puede ser almacenada en memoria para transmisión a un decodificador al momento de la solicitud. La corriente de bits también puede ser transmitida y/o difundida a una pluralidad de decodificadores. La creación de la corriente de bits es un proceso iterativo. Por consiguiente, los pasos 101, 103, 105, 107, y 109 pueden ocurrir de manera continua y/o simultánea sobre muchos cuadros y bloques. El orden mostrado en la figura 1 es presentado por claridad y facilidad de análisis, y no pretende limitar el proceso de codificación de vídeo a un orden particular.

El decodificador recibe la corriente de bits y comienza el proceso de decodificación en el paso 111. Específicamente, el decodificador emplea un esquema de decodificación entrópica para convertir la corriente de bits en datos sintaxis y vídeo correspondientes. El decodificador emplea los datos de sintaxis de la corriente de bits para determinar las divisiones para los cuadros en el paso 111. La división debiera emparejarse con los resultados de la división de bloque en el paso 103. Ahora se describirá la codificación/decodificación entrópica, tal como se emplea en el paso 111. El codificador realiza muchas elecciones durante el proceso de compresión, tal como la selección de esquemas de división de bloque entre varias posibles elecciones con base en el posicionamiento espacial de valores en las

imágenes de entrada. La señalización de las elecciones exactas puede emplear un número grande de bins. Tal como aquí se utiliza, un bin es un valor binario que es tratado como una variable (por ejemplo, un valor de bit que puede variar dependiendo del contexto). La codificación entrópica permite al codificador desechar cualesquiera opciones que claramente no sean viables para un caso particular, dejando un conjunto de opciones permitidas. A cada opción permitida entonces se le asigna una palabra código. La longitud de las palabras código se basa en el número de opciones permitidas (por ejemplo, un bin para dos opciones, dos bins para tres a cuatro opciones, etc.) El codificador entonces codifica la palabra código para la opción seleccionada. Este esquema reduce el tamaño de las palabras código debido a que las palabras código son tan grandes como se desee para indicar de manera única una selección de entre un pequeño sub-conjunto de opciones permitidas en oposición a indicar de manera única la selección de entre un conjunto potencialmente grande de todas las posibles opciones. El decodificador entonces decodifica la selección determinando el conjunto de opciones permitidas en una manera similar al codificador. Al determinar el conjunto de opciones permitidas, el decodificador puede leer la palabra código y puede determinar la selección realizada por el codificador.

En el paso 113, el decodificador ejecuta decodificación de bloque. Específicamente, el decodificador emplea transformadas inversas para generar bloques residuales. Después el decodificador emplea los bloques residuales y bloques de predicción correspondientes para reconstruir los bloques de imagen de acuerdo con la división. Los bloques de predicción pueden incluir tanto bloques de intra-predicción como bloques de inter-predicción tal como son generados en el codificador en el paso 105. Los bloques de imagen reconstruidos entonces son colocados en cuadros de una señal de vídeo reconstruida de acuerdo con los datos de división determinados en el paso 111. La sintaxis para el paso 113 también puede ser señalizada en la corriente de bits a través de codificación entrópica tal como se analizó antes.

En el paso 115, la filtración es ejecutada en los cuadros de la señal de vídeo reconstruida en una manera similar al paso 107 en el codificador. Por ejemplo, filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptativo, y filtros SAO pueden ser aplicados a los cuadros para remover artefactos de bloqueo. Una vez que los cuadros son filtrados, la señal de vídeo puede ser emitida a una pantalla en el paso 117 para visualización por un usuario final.

La figura 2 es un diagrama esquemático de un sistema de codificación y decodificación (códec) ejemplar 200 para codificación de vídeo. Específicamente, el sistema de códec 200 proporciona funcionalidad para soportar la implementación del método operativo 100. El sistema de códec 200 es generalizado para mostrar componentes empleados tanto en un codificador como en un decodificador. El sistema de códec 200 recibe y divide una señal de vídeo tal como se analizó con respecto a los pasos 101 y 103 en el método operativo 100, lo cual tiene como resultado una señal de vídeo dividida 201. El sistema de códec 200 entonces comprime la señal de vídeo dividida 201 en una corriente de bits codificada cuando actúa como un codificador como se analizó con respecto a los pasos 105, 107, y 109 en el método 100. Cuando actúa como un decodificador, el sistema de códec 200 genera una señal de vídeo de salida a partir de la corriente de bits como se analizó con respecto a los pasos 111, 113, 115, y 117 en el método operativo 100. El sistema de códec 200 incluye un componente de control de codificador general 211, un componente de cuantificación y escalación de transformada 213, un componente de estimación de intra-imagen 215, un componente de predicción de intra-imagen 217, un componente de compensación de movimiento 219, un componente de estimación de movimiento 221, un componente de transformada inversa y escalación 229, un componente de análisis de control de filtro 227, un componente de filtros en-bucle 225, un componente de memoria tampón de imágenes decodificadas 223, y un componente de codificación aritmética binaria de contexto adaptativo (CABAC) y formateo de cabecera 231. Dichos componentes son acoplados como se muestra. En la figura 2, líneas negras indican el movimiento de datos que van a ser codificados/decodificados mientras que líneas con guiones indican el movimiento de datos de control que controlan la operación de otros componentes. Los componentes del sistema de códec 200 pueden estar presentes en el codificador. El decodificador puede incluir un subconjunto de los componentes del sistema de códec 200. Por ejemplo, el decodificador puede incluir el componente de predicción de intra-imagen 217, el componente de compensación de movimiento 219, el componente de transformada inversa y escalación 229, el componente de filtros en-bucle 225, y el componente de memoria tampón de imágenes decodificadas 223. Ahora se describirán estos componentes.

La señal de vídeo dividida 201 es una secuencia de vídeo capturada que ha sido dividida en bloques de píxeles por un árbol de codificación. Un árbol de codificación emplea varios modos de división para subdividir un bloque de píxeles en bloques más pequeños de píxeles. Estos bloques entonces pueden ser subdivididos adicionalmente en bloques más pequeños. Los bloques pueden ser referidos como nodos en el árbol de codificación. Nodos padre más grandes son divididos en nodos hijo más pequeños. El número de veces que un nodo es subdividido se refiere como la profundidad del nodo/árbol de codificación. Los bloques divididos pueden ser incluidos en unidades de codificación (CUs) en algunos casos. Por ejemplo, una CU puede ser una sub-porción de una CTU que contiene un bloque de luminancia, bloques de crominancia de diferencia roja (Cr), y bloques de crominancia de diferencia azul (Cb) junto con instrucciones de sintaxis correspondientes para la CU. Los modos de división pueden incluir un árbol binario (BT), árbol triple (TT), y un árbol cuaternario (QT) empleados para dividir un nodo en dos, tres o cuatro nodos

hijo, respectivamente, de diversas formas dependiendo de los modos de división empleados. La señal de vídeo dividida 201 es reenviada al componente de control de codificador general 211, el componente de cuantificación y escalación de transformada 213, el componente de estimación de intra-imagen 215, el componente de análisis de control de filtro 227, y el componente de estimación de movimiento 221 para compresión.

El componente de control de codificador general 211 está configurado para tomar decisiones relacionadas con la codificación de las imágenes de la secuencia de vídeo en la corriente de bits de acuerdo con restricciones de aplicación. Por ejemplo, el componente de control de codificador general 211 gestiona la optimización del tamaño de tasa de bits/corriente de bits versus la calidad de la reconstrucción. Dichas decisiones pueden realizarse con base en el espacio de almacenamiento/disponibilidad de ancho de banda así como las solicitudes de resolución de imagen. El componente de control de codificador general 211 también gestiona la utilización de memoria tampón en virtud de la velocidad de transmisión para mitigar los problemas de empotramiento y rebasamiento de memoria tampón. Para gestionar estos problemas, el componente de control de codificador general 211 gestiona la división, predicción, y filtración por los otros componentes. Por ejemplo, el componente de control de codificador general 211 puede incrementar dinámicamente la complejidad de la compresión para incrementar la resolución e incrementar el uso de ancho de banda o reducir la complejidad de la compresión para disminuir la resolución y el uso de ancho de banda. Por lo tanto, el componente de control de codificador general 211 controla los otros componentes del sistema de códec 200 para equilibrar la calidad de reconstrucción de señal de vídeo con las preocupaciones de tasa de bits. El componente de control de codificador general 211 crea datos de control los cuales controlan la operación de los otros componentes. Los datos de control también son reenviados al componente CABAC y de formateo de cabecera 231 para ser codificados en la corriente de bits con la finalidad de señalar parámetros para decodificación en el decodificador.

La señal de vídeo dividida 201 también es enviada al componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 para inter-predicción. Un cuadro o segmento de la señal de vídeo dividida 201 puede ser dividido en múltiples bloques de vídeo. El componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 ejecutan codificación inter-predictiva del bloque de vídeo recibido con relación a uno o más bloques en uno o más cuadros de referencia para proporcionar predicción temporal. El sistema de códec 200 puede ejecutar múltiples pasos de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

El componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 pueden estar altamente integrados, pero se ilustran separadamente para propósitos conceptuales. La estimación de movimiento, ejecutada por el componente de estimación de movimiento 221, es el proceso de generar vectores de movimiento, los cuales estiman movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un objeto codificado con relación a un bloque predictivo. Un bloque predictivo es un bloque que resulta estar estrechamente emparejado con el bloque que va a ser codificado, en términos de diferencia de pixel. Un bloque predictivo también puede ser referido como un bloque de referencia. Dicha diferencia de pixel puede ser determinada por la suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD), u otras métricas de diferencia. HEVC emplea varios objetos codificados incluyendo una CTU, bloques de árbol de codificación (CTBs), y CUs. Por ejemplo, una CTU puede ser dividida en CTBs, los cuales entonces pueden ser divididos en CBs para inclusión en CUs. Una CU puede ser codificada como una unidad de predicción (PU) que contiene datos de predicción y/o una unidad de transformada (TU) que contiene datos residuales transformados para la CU. El componente de estimación de movimiento 221 genera vectores de movimiento, PUs, y TUs utilizando un análisis de distorsión de tasa como parte de un proceso de optimización de distorsión de tasa. Por ejemplo, el componente de estimación de movimiento 221 puede determinar múltiples bloques de referencia, múltiples vectores de movimiento, etc. para un bloque/cuadro actual, y puede seleccionar los bloques de referencia, vectores de movimiento, etc. teniendo las mejores características de distorsión de tasa. Las mejores características de distorsión de tasa equilibran tanto la reconstrucción de la calidad de vídeo (por ejemplo, cantidad de pérdida de datos por compresión) como la eficiencia de la codificación (por ejemplo, tamaño de la codificación final).

En algunos ejemplos, el sistema de códec 200 puede calcular valores para posiciones de pixel de sub-entero de imágenes de referencia almacenadas en el componente de memoria tampón de imágenes decodificadas 223. Por ejemplo, el sistema de códec de vídeo 200 puede interpolar valores de posiciones de pixel de un-cuarto, posiciones de pixel de un-octavo, u otras posiciones de pixel fraccionales de la imagen de referencia. Por lo tanto, el componente de estimación de movimiento 221 puede ejecutar búsqueda de movimiento con relación a las posiciones de pixel completo y posiciones de pixel fraccional y emitir un vector de movimiento con precisión de pixel fraccional. El componente de estimación de movimiento 221 calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento inter-codificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. El componente de estimación de movimiento 221 emite el vector de movimiento calculado como datos de movimiento al componente CABAC y de formateo de cabecera 231 para codificación y movimiento al componente de compensación de movimiento 219.

La compensación de movimiento, ejecutada por el componente de compensación de movimiento 219, puede involucrar la extracción o generación del bloque predictivo con base en el vector de movimiento determinado por el componente de estimación de movimiento 221. Una vez más, el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 pueden estar funcionalmente integrados, en algunos ejemplos. Al momento de recibir el vector de movimiento para la PU del bloque de vídeo actual, el componente de compensación de movimiento 219 puede localizar el bloque predictivo al cual apunta el vector de movimiento. Un bloque de vídeo residual entonces es formado sustrayendo valores de pixel del bloque predictivo a partir de los valores de pixel del bloque de vídeo actual que está siendo codificado, formando valores de diferencia de pixel. En general, el componente de estimación de movimiento 221 ejecuta estimación de movimiento con relación a los componentes de luminancia, y el componente de compensación de movimiento 219 utiliza vectores de movimiento calculados con base en los componentes de luminancia para componentes de crominancia y componentes de luminancia. El bloque predictivo y el bloque residual son reenviados al componente de cuantificación y escalación de transformada 213.

La señal de vídeo dividida 201 también es enviada al componente de estimación de intra-imagen 215 y al componente de predicción de intra-imagen 217. Al igual que con el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219, el componente de estimación de intra-imagen 215 y el componente de predicción de intra-imagen 217 pueden estar altamente integrados, pero se ilustran separadamente para propósitos conceptuales. El componente de estimación de intra-imagen 215 y el componente de predicción de intra-imagen 217 intra-predicen un bloque actual con relación a bloques en un cuadro actual, como una alternativa a la inter-predicción ejecutada por el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 entre cuadros, como se describió antes. En particular, el componente de estimación de intra-imagen 215 determina un modo de intra-predicción a utilizar para codificar un bloque actual. En algunos ejemplos, el componente de estimación de intra-imagen 215 selecciona un modo de intra-predicción apropiado para codificar un bloque actual a partir de múltiples modos de intra-predicción probados. Los modos de intra-predicción seleccionados son entonces reenviados al componente CABAC y de formateo de cabecera 231 para codificación.

Por ejemplo, el componente de estimación de intra-imagen 215 calcula valores de distorsión de tasa utilizando un análisis distorsión de tasa para los diversos modos de intra-predicción probados, y selecciona el modo de intra-predicción que tenga las mejores características de distorsión de tasa entre los modos probados. El análisis de distorsión de tasa generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque no codificado original que fue codificado para producir el bloque codificado, así como una tasa de bits (por ejemplo, un número de bits) utilizada para producir el bloque codificado. El componente de estimación de intra-imagen 215 calcula relaciones de las distorsiones y tasas para los diversos bloques codificados a fin de determinar cuál modo de intra-predicción exhibe el mejor valor de distorsión de tasa para el bloque. Además, el componente de estimación de intra-imagen 215 puede ser configurado para codificar bloques de profundidad de un mapa de profundidad utilizando un modo de modelado de profundidad (DMM) basado en optimización de distorsión de tasa (RDO).

El componente de predicción de intra-imagen 217 puede generar un bloque residual a partir del bloque predictivo con base en los modos de intra-predicción seleccionados determinados por el componente de estimación de intra-imagen 215 cuando se implementa en un codificador o puede leer el bloque residual de la corriente de bits cuando se implementa en un decodificador. El bloque residual incluye la diferencia en valores entre el bloque predictivo y el bloque original, representado como una matriz. El bloque residual entonces es reenviado al componente de cuantificación y escalación de transformada 213. El componente de estimación de intra-imagen 215 y el componente de predicción de intra-imagen 217 pueden operar en componentes de luminancia y crominancia.

El componente de cuantificación y escalación de transformada 213 está configurado para comprimir adicionalmente el bloque residual. El componente de cuantificación y escalación de transformada 213 aplica una transformada, tal como una transformada discreta del coseno (DCT), una transformada discreta de seno (DST), o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformada residuales. También se podrían utilizar transformadas de ondícula, transformadas de entero, transformadas de sub-banda u otros tipos de transformadas. La transformada puede convertir la información residual de un dominio de valor de pixel en un dominio de transformada, tal como un dominio de frecuencia. El componente de cuantificación y escalación de transformada 213 también está configurado para escalar la información residual transformada, por ejemplo basada en frecuencia. Dicha escalación involucre aplicar un factor de escala a la información residual de manera que diferente información de frecuencia es cuantificada a diferentes granularidades, lo cual puede afectar la calidad visual final del vídeo reconstruido. El componente de cuantificación y escalación de transformada 213 también está configurado para cuantificar los coeficientes de transformada para reducir adicionalmente la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede ser modificado al ajustar un parámetro de cuantificación. En algunos ejemplos, el componente de cuantificación y escalación de transformada 213 entonces puede ejecutar un escaneo de la matriz incluyendo los coeficientes de transformada cuantificados. Los coeficientes

de transformada cuantificados son reenviados al componente CABAC y de formateo de cabecera 231 para ser codificados en la corriente de bits.

El componente de transformada inversa y escalación 229 aplica una operación inversa del componente de cuantificación y escalación de transformada 213 para soportar estimación de movimiento. El componente de transformada inversa y escalación 229 aplica escalación, transformación y/o cuantificación inversa para reconstruir el bloque residual en el dominio de pixel, por ejemplo, para posterior uso como un bloque de referencia el cual se puede convertir en un bloque predictivo para otro bloque actual. El componente de estimación de movimiento 221 y/o el componente de compensación de movimiento 219 pueden calcular un bloque de referencia agregando el bloque residual de regreso a un bloque predictivo correspondiente para uso en estimación de movimiento de un bloque/cuadro posterior. Se aplican filtros a los bloques de referencia reconstruidos para mitigar artefactos creados durante la escalación, cuantificación y transformada. Dichos artefactos de otra manera podrían ocasionar una predicción imprecisa (y crear artefactos adicionales) cuando se predicen bloques subsiguientes.

El componente de análisis de control de filtro 227 y el componente de filtros en-bucle 225 aplican los filtros a los bloques residuales y/o a bloques de imagen reconstruidos. Por ejemplo, el bloque residual transformado del componente de transformada inversa y escalación 229 se puede combinar con un bloque de predicción correspondiente del componente de predicción de intra-imagen 217 y/o componente de compensación de movimiento 219 para reconstruir el bloque de imagen original. Los filtros pueden entonces ser aplicados al bloque de imagen reconstruido. En algunos ejemplos, los filtros en su lugar pueden ser aplicados a los bloques residuales. Al igual que con otros componentes en la figura 2, el componente de análisis de control de filtro 227 y el componente de filtros en-bucle 225 están altamente integrados y pueden ser implementados juntos, pero se muestran separadamente para propósitos conceptuales. Los filtros aplicados a los bloques de referencia reconstruidos son aplicados a regiones espaciales particulares e incluyen múltiples parámetros para ajustar cómo se aplican dichos filtros. El componente de análisis de control de filtro 227 analiza los bloques de referencia reconstruidos para determinar en dónde debieran aplicarse dichos filtros y establece parámetros correspondientes. Dichos datos son reenviados al componente CABAC y de formateo de cabecera 231 como datos de control de filtro para codificación. El componente de filtros en-bucle 225 aplica dichos filtros con base en los datos de control de filtro. Los filtros pueden incluir un filtro de desbloqueo, un filtro de supresión de ruido, un filtro SAO, y un filtro de bucle adaptativo. Dichos filtros pueden ser aplicados en el dominio de pixel/espacial (por ejemplo, en un bloque de píxeles reconstruido) o en el dominio de frecuencia, dependiendo del ejemplo.

Cuando opera como un codificador, el bloque de imagen reconstruido filtrado, el bloque residual, y/o el bloque de predicción son almacenados en el componente de memoria tampón de imágenes decodificadas 223 para posterior uso en estimación de movimiento como se analizó antes. Cuando opera como un decodificador, el componente de memoria tampón de imágenes decodificadas 223 almacena y reenvía los bloques reconstruidos y filtrados hacia una pantalla como parte de una señal de vídeo de salida. El componente de memoria tampón de imágenes decodificadas 223 puede ser cualquier dispositivo de memoria con la capacidad para almacenar bloques de predicción, bloques residuales, y/o bloques de imagen reconstruidos.

El componente CABAC y de formateo de cabecera 231 recibe los datos desde los diversos componentes del sistema de códec 200 y codifica dichos datos en una corriente de bits codificada para transmisión a un decodificador. Específicamente, el componente CABAC y de formateo de cabecera 231 genera varias cabeceras para codificar datos de control, tal como datos de control generales y datos de control de filtro. Además, datos de predicción, incluyendo datos de intra-predicción y movimiento, así como datos residuales en la forma de datos de coeficiente de transformada cuantificados son codificados en la corriente de bits. La corriente de bits final incluye toda la información deseada por el decodificador para reconstruir la señal de vídeo dividida original 201. Dicha información también puede incluir tablas de índice de modo de intra-predicción (también referidas como tablas de mapeo de palabra código), definiciones de contextos de codificación para diversos bloques, indicaciones de modos de intra-predicción más probables, una indicación de información de división, etc. Dichos datos pueden ser codificados empleando codificación entrópica. Por ejemplo, la información puede ser codificada empleando codificación de longitud variable adaptable al contexto (CAVLC), CABAC, codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), codificación entrópica de división por intervalo de probabilidad (PIPE), u otra técnica de codificación entrópica. Después de la codificación entrópica, la corriente de bits codificada puede ser transmitida a otro dispositivo (por ejemplo, un decodificador de vídeo) o se puede lograr para posterior transmisión o recuperación.

La figura 3 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar 300. El codificador de vídeo 300 puede ser empleado para implementar las funciones de codificación del sistema de códec 200 y/o implementar los pasos 101, 103, 105, 107, y/o 109 del método operativo 100. El codificador 300 divide una señal de vídeo de entrada, teniendo como resultado una señal de vídeo dividida 301, la cual es sustancialmente similar a la señal de vídeo dividida 201. La señal de vídeo dividida 301 entonces es comprimida y codificada en una corriente de bits a través de componentes del codificador 300.

Específicamente, la señal de vídeo dividida 301 es reenviada a un componente de predicción de intra-imagen 317 para intra-predicción. El componente de predicción de intra-imagen 317 puede ser sustancialmente similar al componente de estimación de intra-imagen 215 y el componente de predicción de intra-imagen 217. La señal de vídeo dividida 301 también es reenviada a un componente de compensación de movimiento 321 para inter-predicción con base en bloques de referencia en un componente de memoria tampón de imágenes decodificadas 323. El componente de compensación de movimiento 321 puede ser sustancialmente similar al componente de estimación de movimiento 221 y el componente de compensación de movimiento 219. Los bloques de predicción y los bloques residuales del componente de predicción de intra-imagen 317 y el componente de compensación de movimiento 321 son reenviados a un componente de transformada y cuantificación 313 para transformar y cuantificar los bloques residuales. El componente de transformada y cuantificación 313 puede ser sustancialmente similar al componente de cuantificación y escalación de transformada 213. Los bloques residuales y los bloques de predicción transformados y cuantificados correspondientes (junto con datos de control asociados) son reenviados a un componente de codificación entrópica 331 para codificación en una corriente de bits. El componente de codificación entrópica 331 puede ser sustancialmente similar al componente CABAC y de formateo de cabecera 231.

Los bloques residuales y/o los bloques de predicción transformados y cuantificados correspondientes también son reenviados desde el componente de transformada y cuantificación 313 a un componente de transformada y cuantificación inversa 329 para reconstrucción en bloques de referencia para uso por el componente de compensación de movimiento 321. El componente de transformada y cuantificación inversa 329 puede ser sustancialmente similar al componente de transformada inversa y escalación 229. Filtros en-bucle en un componente de filtros en-bucle 325 también son aplicados a los bloques residuales y/o bloques de referencia reconstruidos, dependiendo del ejemplo. El componente de filtros en-bucle 325 puede ser sustancialmente similar al componente de análisis de control de filtro 227 y el componente de filtros en-bucle 225. El componente de filtros en-bucle 325 puede incluir múltiples filtros como se analizó con respecto al componente de filtros en-bucle 225. Los bloques filtrados entonces son almacenados en un componente de memoria tampón de imágenes decodificadas 323 para uso como bloques de referencia por el componente de compensación de movimiento 321. El componente de memoria tampón de imágenes decodificadas 323 puede ser sustancialmente similar al componente de memoria tampón de imágenes decodificadas 223.

La figura 4 es un diagrama de bloques que ilustra un decodificador de vídeo ejemplar 400. El decodificador de vídeo 400 puede ser empleado para implementar las funciones de decodificación del sistema de códec 200 y/o implementar los pasos 111, 113, 115, y/o 117 del método operativo 100. El decodificador 400 recibe una corriente de bits, por ejemplo, desde un codificador 300, y genera una señal de vídeo de salida reconstruida con base en la corriente de bits para despliegue a un usuario final.

La corriente de bits es recibida por un componente de decodificación entrópica 433. El componente de decodificación entrópica 433 está configurado para implementar un esquema de decodificación entrópica, tal como codificación CAVLC, CABAC, SBAC, PIPE, u otras técnicas de codificación entrópica. Por ejemplo, el componente de decodificación entrópica 433 puede emplear información de cabecera para proporcionar un contexto para interpretar datos adicionales codificados como palabras código en la corriente de bits. La información decodificada incluye cualquier información deseada para decodificar la señal de vídeo, dichos datos de control general, datos de control de filtro, información de división, datos de movimiento, datos de predicción, y coeficientes de transformada cuantificados de bloques residuales. Los coeficientes de transformada cuantificados son reenviados a un componente de transformada y cuantificación inversa 429 para reconstrucción en bloques residuales. El componente de transformada y cuantificación inversa 429 puede ser similar al componente de transformada y cuantificación inversa 329.

Los bloques residuales y/o bloques de predicción reconstruidos son reenviados al componente de predicción de intra-imagen 417 para reconstrucción en bloques de imagen con base en operaciones de intra-predicción. El componente de predicción de intra-imagen 417 puede ser similar al componente de estimación de intra-imagen 215 y un componente de predicción de intra-imagen 217. Específicamente, el componente de predicción de intra-imagen 417 emplea modos de predicción para localizar un bloque de referencia en el cuadro y aplica un bloque residual al resultado para reconstruir bloques de imagen intra-predichos. Los bloques de imagen intra-predichos reconstruidos y/o los bloques residuales y los datos de inter-predicción correspondientes son reenviados a un componente de memoria tampón de imágenes decodificadas 423 a través de un componente de filtros en-bucle 425, el cual puede ser sustancialmente similar al componente de memoria tampón de imágenes decodificadas 223 y al componente de filtros en-bucle 225, respectivamente. El componente de filtros en-bucle 425 filtra los bloques de imagen reconstruidos, bloques residuales y/o bloques de predicción, y dicha información es almacenada en el componente de memoria tampón de imágenes decodificadas 423. Bloques de imagen reconstruidos del componente de memoria tampón de imágenes decodificadas 423 son reenviados a un componente de compensación de movimiento 421 para inter-predicción. El componente de compensación de movimiento 421 puede ser sustancialmente similar al componente de estimación de movimiento 221 y/o al componente de compensación de movimiento 219.

Específicamente, el componente de compensación de movimiento 421 emplea vectores de movimiento de un bloque de referencia para generar un bloque de predicción y aplica un bloque residual al resultado para reconstruir un bloque de imagen. Los bloques reconstruidos resultantes también pueden ser reenviados a través del componente de filtros en-bucle 425 al componente de memoria tampón de imágenes decodificadas 423. El componente de memoria tampón de imágenes decodificadas 423 continúa almacenando bloques de imagen reconstruidos adicionales, los cuales pueden ser reconstruidos en cuadros a través de la información de división. Dichos cuadros también pueden ser colocados en una secuencia. La secuencia es emitida hacia una pantalla como una señal de vídeo de salida reconstruida.

La figura 5A es un diagrama esquemático que ilustra una imagen ejemplar 500 dividida en sub-imágenes 510. Por ejemplo, la imagen 500 puede ser dividida para codificación por un sistema de códec 200 y/o un codificador 300 y dividida para decodificación por un sistema de códec 200 y/o un decodificador 400. Como otro ejemplo, la imagen 500 puede ser dividida por un codificador en el paso 103 del método 100 para uso por un decodificador en el paso 111.

Una imagen 500 es una imagen que muestra la porción visual completa de una secuencia de vídeo en una ubicación temporal especificada. La imagen 500 también puede ser referida como una fotografía y/o un cuadro. Una imagen 500 puede ser especificada por un conteo de orden de imágenes (POC). El POC es un índice que indica el orden de salida/despliegue de las imágenes 500 en una secuencia de vídeo. La imagen 500 puede ser dividida en sub-imágenes 510. Una sub-imagen 510 es una región rectangular o cuadrada de uno o más segmentos/grupos de mosaicos dentro de una imagen 500. Sub-imágenes 510 son opcionales, y por lo tanto algunas secuencias de vídeo contienen sub-imágenes 510 mientras que otras no. Aunque se muestran cuatro sub-imágenes 510, la imagen 500 puede ser dividida en cualquier número de sub-imágenes 510. La división de la sub-imagen 510 puede ser consistente sobre toda una secuencia de vídeo codificada (CVS).

Se pueden emplear sub-imágenes 510 para permitir que diferentes regiones de una imagen 500 sean tratadas de manera diferente. Por ejemplo, una sub-imagen especificada 510 puede ser independientemente extraída y transmitida a un decodificador. Como un ejemplo específico, un usuario que emplea un casco de realidad virtual (VR) puede ver un sub-conjunto de la imagen 500, lo cual puede proporcionar al usuario la impresión de estar físicamente presente en un espacio como se muestra en la imagen 500. En dicho caso, la transmisión de solo las sub-imágenes 510 que probablemente van a ser desplegadas al usuario puede incrementar la eficiencia de la codificación. Como otro ejemplo, diferentes sub-imágenes 510 pueden ser tratadas de manera diferente en algunas aplicaciones. Como un ejemplo específico, una aplicación de teleconferencia puede desplegar un altavoz activo a mayor resolución en una posición más prominente que los usuarios que actualmente no están hablando. La colocación de diferentes usuarios en diferentes sub-imágenes 510 soporta la reconfiguración en tiempo real de la imagen desplegada para soportar esta funcionalidad.

Cada sub-imagen 510 puede ser identificada por un ID único de sub-imagen, el cual puede ser consistente para CVS completo. Por ejemplo, la sub-imagen 510 en la parte superior izquierda de la imagen 500 puede tener un ID de sub-imagen de cero. En dicho caso, la sub-imagen superior izquierda 510 de cualquier imagen 500 en la secuencia puede ser referida por el ID de sub-imagen de cero. Además, cada sub-imagen 510 puede incluir una configuración definida, la cual puede ser consistente para todo el CVS. Por ejemplo, una sub-imagen 510 puede contener una altura, una anchura y/o una compensación. La altura y anchura describen el tamaño de la sub-imagen 510 y la compensación describe la ubicación de la sub-imagen 510. Por ejemplo, la suma de las anchuras de todas las sub-imágenes 510 en una fila es la anchura de la imagen 500. Además, la suma de las Alturas de todas las sub-imágenes 510 en una columna es la altura de la imagen 500. Además, la compensación indica la posición de la esquina superior izquierda de la sub-imagen 510 con relación a la esquina superior izquierda de la imagen 500. La altura, anchura y compensación de una sub-imagen 510 proporcionan información suficiente para colocar la sub-imagen correspondiente 510 en la imagen 500. Debido a que la división de las sub-imágenes 510 puede ser consistente sobre un CVS completo, los parámetros relacionados con las sub-imágenes pueden quedar contenidos en un conjunto de parámetros de secuencia (SPS).

La figura 5B es un diagrama esquemático que ilustra una sub-imagen ejemplar 510 dividida en segmentos 515. Como se muestra, la sub-imagen 510 de una imagen 500 puede contener uno o más segmentos 515. Un segmento 515 es un número entero de mosaicos completos o un número entero de filas de CTU completas consecutivas dentro de un mosaico de una imagen que están exclusivamente contenidos en una sola unidad de capa de abstracción de red (NAL). Aunque se muestran cuatro segmentos 515, la sub-imagen 510 puede incluir cualquier número de segmentos 515. Los segmentos 515 contienen datos visuales que son específicos de una imagen 500 de un POC especificado. Por consiguiente, parámetros relacionados con segmentos 515 pueden estar contenidos en un conjunto de parámetros de imagen (PPS) y/o una cabecera de segmento.

La figura 5C es un diagrama esquemático que ilustra un segmento ejemplar 515 dividido en mosaicos 517. Tal como se muestra, el mosaico 515 de una imagen 500 puede contener uno o más mosaicos 517. Los mosaicos 517 pueden

ser creados dividiendo la imagen 500 en filas y columnas de rectángulos. Por consiguiente, un mosaico 517 es una región rectangular o cuadrada de CTUs dentro de una columna de mosaicos particular y una fila de mosaicos particular en una imagen. La formación de mosaicos es opcional, y por lo tanto algunas secuencias de vídeo contienen los mosaicos 517 mientras que otras no. Aunque se muestran cuatro mosaicos 517, el segmento 515 puede incluir cualquier número de mosaicos 517. Los mosaicos 517 pueden contener datos visuales que son específicos de un segmento 515 de una imagen 500 de un POC especificado. En algunos casos, los segmentos 515 también pueden estar contenidos en los mosaicos 517. Por consiguiente, parámetros relacionados con los mosaicos 517 pueden estar contenidos en un PPS y/o una cabecera de segmento.

La figura 5D es un diagrama esquemático que ilustra un segmento ejemplar 515 dividido en CTUs 519. Como se muestra, el segmento 515 (o un mosaico 517 de un segmento 515) de una imagen 500 puede contener una o más CTUs 519. Una CTU 519 es una región de la imagen 500 que es sub-dividida por un árbol de codificación para crear bloques de codificación, los cuales son codificados/decodificados. Una CTU 519 puede contener muestras de luminancia para una imagen monocromática 500 o una combinación de muestras de luminancia y crominancia para una imagen de color 500. Un agrupamiento de muestras de luminancia o muestras de crominancia que pueden ser divididas por un árbol de codificación es referido como un bloque de árbol de codificación (CTB) 518. Debido a esto, una CTU 519 contiene un CTB 518 de muestras de luminancia y dos CTBs 518 correspondientes de muestras de crominancia de una imagen 500 que tiene tres arreglos de muestra, o contiene un CTB 518 de muestras de una imagen monocromática o una imagen que es codificada utilizando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras.

Tal como se mostró antes, una imagen 500 puede ser dividida en sub-imágenes 510, segmentos 515, mosaicos 517, CTUs 519, y/o CTBs 518, los cuales entonces son divididos en bloques. Dichos bloques son entonces codificados para transmisión a un decodificador. La decodificación de dichos bloques puede tener como resultado una imagen decodificada que contiene varios tipos de ruido. Para corregir dichos problemas, sistemas de codificación de vídeo pueden aplicar varios filtros a través de límites de bloque. Estos filtros pueden remover el bloqueo, ruido de cuantificación, y otros artefactos de codificación indeseables. Como se observe antes, las sub-imágenes 510 pueden ser empleadas cuando se ejecuta extracción independiente. En este caso, una sub-imagen actual 510 puede ser decodificada y desplegada sin decodificar información de otras sub-imágenes 510. Debido a esto, límites de bloque a lo largo del borde de sub-imagen 510 se pueden alinear con límites de sub-imagen. En algunos casos, los límites de bloque también se pueden alinear con límites de mosaico. Se pueden aplicar filtros a través de dichos límites de bloque, y por lo tanto se pueden aplicar a través de límites de sub-imagen y/o límites de mosaico. Esto puede provocar errores cuando una sub-imagen actual 510 es extraída independientemente debido a que el proceso de filtración puede operar en una manera inesperada cuando los datos de una sub-imagen adyacente 510 no están disponibles.

A fin de abordar estos problemas, se puede emplear un indicador que controle la filtración al nivel de sub-imagen 510. Por ejemplo, el indicador puede ser denotado como un `loop_filter_across_subpic_enabled_flag`. Cuando se establece el indicador para una sub-imagen 510, se pueden aplicar filtros a través del límite de sub-imagen correspondiente. Cuando no se establece el indicador, los filtros no son aplicados a través del límite de sub-imagen correspondiente. De esta manera, los filtros pueden ser apagados para sub-imágenes 510 que son codificadas para extracción separada o encendidos para sub-imágenes 510 que son codificadas para despliegue como un grupo. Se puede establecer otro indicador para controlar la filtración al nivel de mosaico 517. El indicador puede ser denotado como un `loop_filter_across_tiles_enabled_flag`. Cuando se establece el indicador para un mosaico 517, se pueden aplicar filtros a través del límite de mosaico. Cuando no se establece el indicador, los filtros no son aplicados a través del límite de mosaico. De esta manera, Los filtros pueden ser apagados o encendidos para uso en límites de mosaico (por ejemplo, mientras se continúan filtrando las porciones internas del mosaico).

También como se observó antes, la formación de mosaicos es opcional. Sin embargo, algunos sistemas de codificación de vídeo describen los límites de sub-imagen en términos de los mosaicos 517 incluidos en la sub-imagen 510. En dichos sistemas, las descripciones de límite de sub-imagen en términos de mosaicos 517 limitan el uso de sub-imágenes 510 a imágenes 500 que emplean mosaicos 517. A fin de ensanchar la aplicabilidad de sub-imágenes 510, las sub-imágenes 510 pueden ser descritas en términos de límites, en términos de CTBs 518, y/o CTUs 519. Específicamente, la anchura y altura de una sub-imagen 510 pueden ser señalizadas en unidades de CTBs 518. Además, la posición de la CTU superior izquierda 519 de la sub-imagen 510 puede ser señalizada como una compensación de la CTU superior izquierda 519 de la imagen 500 tal como se mide en CTBs 518. Tamaños de CTU 519 y CTB 518 pueden ser establecidos a un valor predeterminado. Por consiguiente, la señalización de las dimensiones y posición de la sub-imagen en términos de CTBs 518 y CTUs 519 proporciona suficiente información para que un decodificador coloque la sub-imagen 510 para despliegue. Esto permite que las sub-imágenes 510 sean empleadas incluso cuando no se emplean los mosaicos 517. Tal como aquí se utiliza, se aplica un filtro a través de un límite de sub-imagen 510 o mosaico 517 cuando el filtro es aplicado a muestras en ambos lados del límite.

Además, algunos sistemas de codificación de vídeo direccionan los segmentos 515 con base en su posición con relación a la imagen 500. Esto crea un problema cuando sub-imágenes 510 son codificadas para extracción y despliegue independientes. En dicho caso, segmentos 515 y direcciones correspondientes asociadas con las sub-imágenes 510 omitidas también se omiten. La omisión de las direcciones de los segmentos 515 puede evitar que el decodificador coloque de manera apropiada los segmentos 515. Algunos sistemas de codificación de vídeo abordan este problema reescribiendo dinámicamente las direcciones en las cabeceras de segmento asociadas con los segmentos 515. Debido a que el usuario puede solicitar cualquier sub-imagen, dicha reescritura ocurre cada vez que un usuario solicita el vídeo, el cual es extremadamente intenso en recursos. Para superar este problema, los segmentos 515 son direccionados con relación a la sub-imagen 510 que contiene el segmento 515 cuando se emplean sub-imágenes 510. Por ejemplo, el segmento 515 puede ser identificado por un índice u otro valor que es específico para la sub-imagen 510 que contiene el segmento 515. La dirección del segmento puede ser codificada en la cabecera de segmento asociada con el segmento 515. El ID de sub-imagen de la sub-imagen 510 que contiene el segmento 515 también se puede codificar en una cabecera de segmento. Además, las dimensiones/configuración de la sub-imagen 510 pueden ser codificadas en el SPS junto con el ID de sub-imagen. Debido a esto, el decodificador puede obtener la configuración de sub-imagen 510 del SPS con base en el ID de sub-imagen y puede colocar el segmento 515 en la sub-imagen 510 sin hacer referencia a la imagen completa 500. Debido a esto, la reescritura de la cabecera de segmento se puede omitir cuando se extrae una sub-imagen 510, lo cual reduce significativamente el uso en el codificador, el decodificador, y/o un segmentador correspondiente.

Una vez que una imagen 500 es dividida en CTBs 518 y/o CTUs 519, los CTBs 518 y/o CTUs 519 pueden ser divididos adicionalmente en bloques de codificación. Los bloques de codificación entonces pueden ser codificados de acuerdo con intra-predicción y/o inter-predicción. La presente divulgación también incluye mejoras relacionadas con los mecanismos de inter-predicción. La inter-predicción puede ser ejecutada en diferentes modos que pueden operar de acuerdo con inter-predicción unidireccional y/o inter-predicción bidireccional.

La figura 6 es un diagrama esquemático que ilustra un ejemplo de inter-predicción unidireccional 600, por ejemplo tal como se ejecuta para determinar vectores de movimiento (MVs) en el paso de compresión de bloque 105, paso de decodificación de bloque 113, componente de estimación de movimiento 221, componente de compensación de movimiento 219, componente de compensación de movimiento 321, y/o componente de compensación de movimiento 421. Por ejemplo, se puede emplear inter-predicción unidireccional 600 para determinar vectores de movimiento para bloques codificados y/o de decodificación creados cuando se divide una imagen, tal como la imagen 500.

La inter-predicción unidireccional 600 emplea un cuadro de referencia 630 con un bloque de referencia 631 para predecir un bloque actual 611 en un cuadro actual 610. El cuadro de referencia 630 puede ser temporalmente colocado después del cuadro actual 610 como se muestra (por ejemplo, como un cuadro de referencia subsiguiente), pero también puede ser temporalmente colocado antes que el cuadro actual 610 (por ejemplo, como un cuadro de referencia precedente) en algunos ejemplos. El cuadro actual 610 es un cuadro/imagen ejemplar que está siendo codificado/decodificado en un tiempo particular. El cuadro actual 610 contiene un objeto en el bloque actual 611 que se empareja con un objeto en el bloque de referencia 631 del cuadro de referencia 630. El cuadro de referencia 630 es un cuadro que es empleado como una referencia para codificar un cuadro actual 610, y un bloque de referencia 631 es un bloque en el cuadro de referencia 630 que contiene un objeto también contenido en el bloque actual 611 del cuadro actual 610.

El bloque actual 611 es cualquier unidad de codificación que está siendo codificada/decodificada en un punto especificado en el proceso de codificación. El bloque actual 611 puede ser un bloque dividido completo, o puede ser un sub-bloque cuando se emplea modo de inter-predicción afín. El cuadro actual 610 es separado del cuadro de referencia 630 por cierta distancia temporal (TD) 633. La TD 633 indica una cantidad de tiempo entre el cuadro actual 610 y cuadro de referencia 630 en una secuencia de vídeo, y puede ser medida en unidades de cuadros. La información de predicción para el bloque actual 611 puede hacer referencia al cuadro de referencia 630 y/o bloque de referencia 631 por un índice de referencia que indica la dirección y distancia temporal entre los cuadros. Durante el periodo de tiempo representado por la TD 633, el objeto en el bloque actual 611 se mueve desde una posición en el cuadro actual 610 a otra posición en el cuadro de referencia 630 (por ejemplo, la posición del bloque de referencia 631). Por ejemplo, el objeto se puede mover a lo largo de una trayectoria de movimiento 613, la cual es una dirección de movimiento de un objeto sobre tiempo. Un vector de movimiento 635 describe la dirección y magnitud del movimiento del objeto a lo largo de la trayectoria de movimiento 613 sobre la TD 633. Por consiguiente, un vector de movimiento codificado 635, un bloque de referencia 631, y un residual incluyendo la diferencia entre el bloque actual 611 y el bloque de referencia 631 proporciona información suficiente para reconstruir un bloque actual 611 y colocar el bloque actual 611 en el cuadro actual 610.

La figura 7 es un diagrama esquemático que ilustra un ejemplo de inter-predicción bidireccional 700, por ejemplo, tal como se ejecuta para determinar MVs en el paso de compresión de bloque 105, el paso de compresión de bloque 113, el componente de estimación de movimiento 221, el componente de compensación de movimiento 219, el

componente de compensación de movimiento 321, y/o el componente de compensación de movimiento 421. Por ejemplo, se puede emplear inter-predicción bidireccional 700 para determinar vectores de movimiento para bloques codificados y/o de decodificación creados cuando se divide una imagen, tal como la imagen 500.

La inter-predicción bidireccional 700 es similar a la inter-predicción unidireccional 600, pero emplea un par de cuadros de referencia para predecir un bloque actual 711 en un cuadro actual 710. Por lo tanto, el cuadro actual 710 y el bloque actual 711 son sustancialmente similares al cuadro actual 610 y el bloque actual 611, respectivamente. El cuadro actual 710 es temporalmente colocado entre un cuadro de referencia precedente 720, el cual ocurre antes que el cuadro actual 710 en la secuencia de vídeo, y un cuadro de referencia subsiguiente 730, el cual ocurre después del cuadro actual 710 en la secuencia de vídeo. El cuadro de referencia precedente 720 y el cuadro de referencia subsiguiente 730 de otra manera son sustancialmente similares al cuadro de referencia 630.

El bloque actual 711 es emparejado con un bloque de referencia precedente 721 en el cuadro de referencia precedente 720 y con un bloque de referencia subsiguiente 731 en el cuadro de referencia subsiguiente 730. Dicho emparejamiento indica que, durante el curso de la secuencia de vídeo, un objeto se mueve desde una posición en el bloque de referencia precedente 721 a una posición en el bloque de referencia subsiguiente 731 a lo largo de una trayectoria de movimiento 713 y a través del bloque actual 711. El cuadro actual 710 está separado del cuadro de referencia precedente 720 por cierta distancia temporal precedente (TD0) 723 y está separado del cuadro de referencia subsiguiente 730 por cierta distancia temporal subsiguiente (TD1) 733. La TD0 723 indica una cantidad de tiempo entre el cuadro de referencia precedente 720 y el cuadro actual 710 en la secuencia de vídeo en unidades de cuadros. La TD1 733 indica una cantidad de tiempo entre el cuadro actual 710 y el cuadro de referencia subsiguiente 730 en la secuencia de vídeo en unidades de cuadro. Por lo tanto, el objeto se mueve del bloque de referencia precedente 721 al bloque actual 711 a lo largo de la trayectoria de movimiento 713 durante un periodo de tiempo indicado por TD0 723. El objeto también se mueve del bloque actual 711 al bloque de referencia subsiguiente 731 a lo largo de la trayectoria de movimiento 713 durante un periodo de tiempo indicado por TD1 733. La información de predicción para el bloque actual 711 puede hacer referencia al cuadro de referencia precedente 720 y/o bloque de referencia precedente 721 y el cuadro de referencia subsiguiente 730 y/o bloque de referencia subsiguiente 731 por un par de índices de referencia indicando la dirección y distancia temporal entre los cuadros.

Un vector de movimiento precedente (MV0) 725 describe la dirección y magnitud del movimiento del objeto a lo largo de la trayectoria de movimiento 713 sobre la TD0 723 (por ejemplo, entre el cuadro de referencia precedente 720 y el cuadro actual 710). Un vector de movimiento subsiguiente (MV1) 735 describe la dirección y magnitud del movimiento del objeto a lo largo de la trayectoria de movimiento 713 sobre la TD1 733 (por ejemplo, entre el cuadro actual 710 y el cuadro de referencia subsiguiente 730). Debido a esto, en inter-predicción bidireccional 700, el bloque actual 711 puede ser codificado y reconstruido empleando el bloque de referencia precedente 721 y/o el bloque de referencia subsiguiente 731, MV0 725, y MV1 735.

En ambos, el modo de fusión y el modo de predicción de vector de movimiento avanzada (AMVP), se genera una lista de candidatos agregando vectores de movimiento candidatos a una lista de candidatos en un orden definido por un patrón de determinación de lista de candidatos. Dichos vectores de movimiento candidatos pueden incluir vectores de movimiento de acuerdo con inter-predicción unidireccional 600, inter-predicción bidireccional 700, o combinaciones de las mismas. Específicamente, se generan vectores de movimiento para bloques vecinos cuando dichos bloques son codificados. Dichos vectores de movimiento son agregados a una lista de candidatos para el bloque actual, y el vector de movimiento para el bloque actual es seleccionado de la lista de candidatos. El vector de movimiento entonces puede ser señalado como el índice del vector de movimiento seleccionado en la lista de candidatos. El decodificador puede construir la lista de candidatos utilizando el mismo proceso que el codificador, y puede determinar el vector de movimiento seleccionado de la lista de candidatos con base en el índice señalado. Por lo tanto, los vectores de movimiento candidatos incluyen vectores de movimiento generados de acuerdo con la inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700, dependiendo de cuál enfoque es utilizado cuando dichos bloques vecinos son codificados.

La figura 8 es un diagrama esquemático que ilustra un ejemplo 800 de la codificación de un bloque actual 801 con base en vectores de movimiento candidatos de bloques codificados adyacentes 802. Un método operativo 100 del codificador 300 y/o un decodificador 400 y/o que emplea la funcionalidad del sistema de códec 200 puede emplear los bloques adyacentes 802 para generar una lista de candidatos. Dicha lista de candidatos puede ser empleada en inter-predicción de acuerdo con una inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700. La lista de candidatos entonces puede ser empleada para codificar/decodificar el bloque actual 801, el cual puede ser generado dividiendo una imagen, tal como la imagen 500.

El bloque actual 801 es un bloque que está siendo codificado en un codificador o decodificado en un decodificador, dependiendo del ejemplo, en un tiempo especificado. Los bloques codificados 802 son bloques que ya están codificados en el tiempo especificado. Por lo tanto, los bloques codificados 802 están potencialmente disponibles para uso cuando se genera una lista de candidatos. El bloque actual 801 y los bloques codificados 802 pueden ser

incluidos en un cuadro común y/o pueden ser incluidos en un cuadro temporalmente adyacente. Cuando los bloques codificados 802 son incluidos en un cuadro común con el bloque actual 801, los bloques codificados 802 contienen un límite que es inmediatamente adyacente a (por ejemplo, topa con) un límite del bloque actual 801. Cuando un bloque codificado 802 es incluido en un cuadro temporalmente adyacente, el bloque codificado 802 está ubicado en la misma posición en el cuadro temporalmente adyacente que la posición del bloque actual 801 en el cuadro actual. La lista de candidatos puede ser generada agregando los vectores de movimiento de los bloques codificados 802 como vectores de movimiento candidatos. El bloque actual 801 entonces puede ser codificado seleccionando un vector de movimiento candidato de la lista de candidatos y señalizando el índice del vector de movimiento candidato seleccionado.

La figura 9 es un diagrama esquemático que ilustra un patrón ejemplar 900 para determinar una lista de candidatos de vectores de movimiento. Específicamente, método operativo 100 de un codificador 300 y/o un decodificador 400 y/o que emplea la funcionalidad del sistema de códec 200 puede emplear el patrón de determinación de lista de candidatos 900 para uso en la generación de una lista de candidatos 911 para la codificación de un bloque actual 801 dividido a partir de una imagen 500. La lista de candidatos resultante 911 puede ser una lista de candidatos de fusión o una lista de candidatos AMVP, las cuales pueden ser empleadas en inter-predicción de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700.

Cuando se codifica un bloque actual 901, el patrón de determinación de lista de candidatos 900 busca las posiciones 905, denotadas como A0, A1, B0, B1, y/o B2, en la misma imagen/cuadro que el bloque actual 901 para vectores de movimiento candidatos válidos. El patrón de determinación de lista de candidatos 900 también puede buscar el bloque colocado 909 para vectores de movimiento candidatos válidos. El bloque colocado 909 es un bloque en la misma posición que el bloque actual 901, pero está incluido en una imagen/cuadro temporalmente adyacente. Los vectores de movimiento candidatos entonces pueden ser colocados en una lista de candidatos 911 en un orden de comprobación predeterminado. Por consiguiente, la lista de candidatos 911 es una lista generada por procedimiento de vectores de movimiento candidatos indexados.

Se puede emplear la lista de candidatos 911 para seleccionar un vector de movimiento para ejecutar inter-predicción para el bloque actual 901. Por ejemplo, el codificador puede obtener las muestras de los bloques de referencia apuntados por los vectores de movimiento candidatos de la lista de candidatos 911. El codificador entonces puede seleccionar el vector de movimiento candidato que apunta al bloque de referencia que se empareja de manera más estrecha con el bloque actual 901. El índice del vector de movimiento candidato seleccionado entonces puede ser codificado para representar el bloque actual 901. En algunos casos, los vectores de movimiento candidatos apuntan a un bloque de referencia que contiene muestras de referencia parciales 915. En este caso, se puede emplear un filtro de interpolación 913 para reconstruir las muestras de referencia completas 915 para soportar la selección de vector de movimiento. Un filtro de interpolación 913 es un filtro con la capacidad de muestrear hacia arriba una señal. Específicamente, el filtro de interpolación 913 es un filtro con la capacidad para aceptar una señal de calidad parcial/inferior como una entrada y determinar una aproximación de una señal de calidad superior/más completa. Debido a esto, el filtro de interpolación 913 puede ser empleado en algunos casos para obtener un conjunto completo de muestras de referencia 915 para uso en la selección del bloque de referencia para el bloque actual 901, y por lo tanto en la selección del vector de movimiento para codificar el bloque actual 901.

Los mecanismos precedentes para codificar un bloque con base en la inter-predicción utilizando una lista de candidatos pueden provocar algunos errores cuando se emplean sub-imágenes, tales como las sub-imágenes 510. Específicamente, los problemas pueden ocurrir cuando un bloque actual 901 está contenido en una sub-imagen actual, pero un vector de movimiento apunta a un bloque de referencia colocado al menos parcialmente en una sub-imagen adyacente. En dicho caso, la sub-imagen actual puede ser extraída para presentación sin la sub-imagen adyacente. Cuando ocurre esto, las porciones del bloque de referencia en la sub-imagen adyacente no pueden ser transmitidas al decodificador, y por lo tanto el bloque de referencia no puede estar disponible para decodificación del bloque actual 901. Cuando esto ocurre, el decodificador no tiene acceso a suficientes datos para decodificar el bloque actual 901.

La presente divulgación proporciona mecanismos para abordar este problema. En un ejemplo, se emplea un indicador que indica que la sub-imagen actual debiera ser tratada como una imagen. Este indicador puede ser establecido para soportar extracción separada de la sub-imagen. Específicamente, cuando se establece el indicador, la sub-imagen actual debiera ser codificada sin hacer referencia a datos en otras sub-imágenes. En este caso, la sub-imagen actual es tratada como una imagen ya que la sub-imagen actual es codificada separadamente de otras sub-imágenes y puede ser desplegada como una imagen separada. Por lo tanto, este indicador puede ser denotado como una `subpic_treated_as_pic_flag[i]` donde *i* es un índice de la sub-imagen actual. Cuando se establece el indicador, los candidatos de vector de movimiento (también conocidos como predictores de vector de movimiento) obtenidos de un bloque colocado 909 incluyen solamente vectores de movimiento que apuntan dentro de la sub-imagen actual. Cualesquiera predictores de vector de movimiento que apuntan fuera de la sub-imagen actual son excluidos de la lista de candidatos 911. Esto garantiza que los vectores de movimiento que apuntan fuera de la sub-

imagen actual no sean seleccionados y se evitan errores asociados. Este ejemplo aplica específicamente a vectores de movimiento del bloque colocado 909. Vectores de movimiento de las posiciones de búsqueda 905 en la misma imagen/cuadro pueden ser corregidos por mecanismos separados como se describe a continuación.

Se puede emplear otro ejemplo para direccionar las posiciones de búsqueda 905 cuando la sub-imagen actual es tratada como una imagen (por ejemplo, cuando se establece el subpic_treated_as_pic_flag[i]). Cuando la sub-imagen actual es tratada como una imagen, la sub-imagen actual debiera ser extraída sin referencia a otras sub-imágenes. El mecanismo ejemplar se relaciona con filtros de interpolación 913. Un filtro de interpolación 913 puede ser aplicado a muestras en una ubicación para interpolar (por ejemplo, predecir) muestras relacionadas en otra ubicación. En el presente ejemplo, vectores de movimiento de bloques codificados en las posiciones de búsqueda 905 pueden apuntar a muestras de referencia 915 fuera de la sub-imagen actual siempre y cuando los filtros de interpolación 913 puedan interpolar dichas muestras de referencia 915 con base únicamente en las muestras de referencia 915 de la sub-imagen actual. Por consiguiente, el presente ejemplo emplea una función de recorte que es aplicada cuando se aplican filtros de interpolación 913 a candidatos de vector de movimiento de las posiciones de búsqueda 905 de la misma imagen. Esta función de recorte recorta datos de las sub-imágenes adyacentes y, por lo tanto, remueve dichos datos como entrada al filtro de interpolación 913 cuando se determinan muestras de referencia 915 apuntadas por candidatos de vector de movimiento. Este enfoque mantiene la separación entre las sub-imágenes durante la codificación para soportar la extracción separada y la decodificación cuando la sub-imagen es tratada como una imagen. La función de recorte puede ser aplicada a un proceso de interpolación bilineal de muestra de luminancia, un proceso de filtración de interpolación de ocho derivaciones de muestra de luminancia, y/o un proceso de interpolación de muestra de crominancia.

La figura 10 es un diagrama de bloques que ilustra un ejemplo de filtro en-bucle 1000. Se puede emplear el filtro en-bucle 1000 para implementar filtros en-bucle 225, 325, y/o 425. Además, el filtro en-bucle 1000 puede ser aplicado en el codificador y el decodificador cuando se ejecuta el método 100. Además, el filtro en-bucle 1000 puede ser aplicado para filtrar un bloque actual 801 dividido a partir de una imagen 500, la cual puede ser codificada de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700 con base en una lista de candidatos generada de acuerdo con el patrón 900. El filtro en-bucle 1000 incluye un filtro de desbloqueo 1043, un filtro SAO 1045, y un filtro de bucle adaptativo (ALF) 1047. Los filtros del filtro en-bucle 1000 son aplicados en secuencia a bloques de imagen reconstruidos en un codificador (por ejemplo, antes de su uso como bloques de referencia) y en el decodificador antes del despliegue.

El filtro de desbloqueo 1043 está configurado para remover bordes formados en bloque creados por inter e intra predicción basada en bloque. El filtro de desbloqueo 1043 escanea una porción de imagen (por ejemplo, un segmento de imagen) para discontinuidades en valores de crominancia y/o luminancia que ocurren en límites de división. El filtro de desbloqueo 1043 entonces aplica una función de suavización a los límites de bloque para remover dichas discontinuidades. La resistencia del filtro de desbloqueo 1043 puede ser modificada dependiendo de la actividad espacial (por ejemplo, varianza de componentes de luminancia/crominancia) que ocurre en un área adyacente a los límites de bloque.

El filtro SAO 1045 está configurado para remover artefactos relacionados con la distorsión de muestra provocada por el proceso de codificación. El filtro SAO 1045 en un codificador clasifica las muestras desbloqueadas de una imagen reconstruida en varias categorías con base en la forma y/o dirección de borde de desbloqueo relativa. Una compensación es entonces determinada y agregada a las muestras con base en las categorías. Las compensaciones son entonces codificadas en la corriente de bits y empleadas por el filtro SAO 1045 en el decodificador. El filtro SAO 1045 remueve los artefactos de formación de bandas (bandas de valores en lugar de transiciones suaves) y artefactos de formación de anillos (señales ficticias cerca de los bordes afilados).

El ALF 1047, en el codificador, está configurado para comparar una imagen reconstruida con una imagen original. El ALF 1047 determina coeficientes que describen las diferencias entre la imagen reconstruida y la imagen original, por ejemplo, a través de un filtro adaptativo basado en Wiener. Dichos coeficientes son codificados en la corriente de bits y empleados por el ALF 1047 en el decodificador para remover las diferencias entre la imagen reconstruida y la imagen original.

Datos de imagen filtrados por el filtro en-bucle 1000 son emitidos a una memoria tampón de imágenes 1023, la cual es sustancialmente similar a la memoria tampón de imágenes decodificadas 223, 323, y/o 423. Tal como se observó antes, el filtro de desbloqueo 1043, el filtro SAO 1045, y/o el ALF 1047 pueden ser apagados en los límites de sub-imagen y/o límites de mosaico por indicadores, tal como un loop_filter_across_subpic_enabled_flag y/o un loop_filter_across_tiles_enabled_flag, respectivamente.

La figura 11 es un diagrama esquemático que ilustra una corriente de bits ejemplar 1100 que contiene parámetros de herramientas de codificación para soportar la decodificación de una sub-imagen de una imagen. Por ejemplo, la corriente de bits 1100 puede ser generada por un sistema de códec 200 y/o un codificador 300 para decodificación

por un sistema de códec 200 y/o un decodificador 400. Como otro ejemplo, la corriente de bits 1100 puede ser generada por un codificador en el paso 109 del método 100 para uso por un decodificador en el paso 111. Además, la corriente de bits 1100 puede contener una imagen codificada 500, sub-imágenes correspondientes 510, y/o bloques codificados asociados, tal como los bloques actuales 801 y/o 901, los cuales pueden ser codificados de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700 con base en una lista de candidatos generados de acuerdo con el patrón 900. La corriente de bits 1100 también puede contener parámetros para configurar el filtro en-bucle 1000.

La corriente de bits 1100 incluye un conjunto de parámetros de secuencia (SPS) 1110, una pluralidad de conjuntos de parámetros de imagen (PPSs) 1111, una pluralidad de cabeceras de segmento 1115, y datos de imagen 1120. Un SPS 1110 contiene datos de secuencia comunes a todas las imágenes en la secuencia de vídeo contenida en la corriente de bits 1100. Dichos datos pueden incluir dimensionamiento de imagen, profundidad de bit, parámetros de herramientas de codificación, restricciones de tasa de bits, etc. El PPS 1111 contiene parámetros que aplican a una imagen completa. Por lo tanto, cada imagen en la secuencia de vídeo se puede referir a un PPS 1111. Se debiera observar que, aunque cada imagen se refiere a un PPS 1111, un solo PPS 1111 puede contener datos para múltiples imágenes en algunos ejemplos. Por ejemplo, múltiples imágenes similares pueden ser codificadas de acuerdo con parámetros similares. En dicho caso, un solo PPS 1111 puede contener datos para dichas imágenes similares. El PPS 1111 puede indicar herramientas de codificación disponibles para segmentos en imágenes correspondientes, parámetros de cuantificación, compensaciones, etc. La cabecera de segmento 1115 contiene parámetros que son específicos para cada segmento en una imagen. Por lo tanto, puede haber una cabecera de segmento 1115 por segmento en la secuencia de vídeo. La cabecera de segmento 1115 puede contener información de tipo de segmento, conteos de orden de imagen (POCs), listas de imágenes de referencia, pesos de predicción, puntos de entrada de mosaico, parámetros de desbloqueo, etc. Se debiera observar que una cabecera de segmento 1115 también puede ser referida como una cabecera de grupo de mosaicos en algunos contextos.

Los datos de imagen 1120 contienen datos de vídeo codificados de acuerdo con inter-predicción y/o intra-predicción así como datos residuales transformados y cuantificados correspondientes. Por ejemplo, una secuencia de vídeo incluye una pluralidad de imágenes codificadas como datos de imagen. Una imagen es un solo cuadro de una secuencia de vídeo y por lo tanto generalmente es desplegada como una sola unidad cuando se despliega la secuencia de vídeo. Sin embargo, se pueden desplegar sub-imágenes para implementar algunas tecnologías tales como realidad virtual, imagen en imagen, etc. Las imágenes hacen referencia a un PPS 1111. Las imágenes son divididas en sub-imágenes, mosaicos, y/o segmentos tal como se analizó antes. En algunos sistemas, los segmentos son referidos como grupos de mosaicos que contienen mosaicos. Los segmentos y/o grupos de mosaicos de los mosaicos hacen referencia a una cabecera de segmento 1115. Los segmentos son divididos adicionalmente en CTUs y/o CTBs. Los CTUs/CTBs son divididos adicionalmente en bloques de codificación con base en los árboles de codificación. Los bloques de codificación entonces pueden ser codificados/decodificados de acuerdo con los mecanismos de predicción.

Los conjuntos de parámetros en la corriente de bits 1100 contienen varios datos que pueden ser empleados para implementar los ejemplos aquí descritos. Para soportar una primera implementación ejemplar, el SPS 1110 de la corriente de bits 1100 contiene una sub-imagen tratada como un indicador de imagen 1131 relacionado con una sub-imagen especificada. En algunos ejemplos, la sub-imagen tratada como un indicador de imagen 1131 es denotada como `unsubpic_treated_as_pic_flag[i]` donde *i* es un índice de la sub-imagen asociada con el indicador. Por ejemplo, la sub-imagen tratada como un indicador de imagen 1131 puede ser establecida igual a uno para especificar que una *i*-ava sub-imagen de cada imagen codificada en una secuencia de vídeo codificada (en los datos de imagen 1120) es tratada como una imagen en un proceso de decodificación exclusivo de operaciones de filtración en-bucle. La sub-imagen tratada como un indicador de imagen 1131 puede ser empleada cuando una sub-imagen actual en una imagen actual ha sido codificada de acuerdo con inter-predicción. Cuando la sub-imagen tratada como un indicador de imagen 1131 es establecida para indicar que la sub-imagen actual es tratada como una imagen, una lista de candidatos de vectores de movimiento candidatos para un bloque actual puede ser determinada excluyendo los vectores de movimiento colocados de la lista de candidatos que están incluidos en un bloque colocado y que apuntan fuera de la sub-imagen actual. Esto garantiza que vectores de movimiento que apuntan fuera de la sub-imagen actual no sean seleccionados y que se eviten errores asociados cuando la sub-imagen actual es extraída separadamente de otras sub-imágenes.

En algunos ejemplos, la lista de candidatos de vectores de movimiento para el bloque actual es determinada de acuerdo con la predicción de vector de movimiento de luminancia temporal. Por ejemplo, la predicción de vector de movimiento de luminancia temporal puede ser empleada cuando el bloque actual es un bloque de luminancia de muestras de luminancia, un vector de movimiento actual seleccionado para el bloque actual es un vector de movimiento de luminancia temporal que apunta a muestras de luminancia de referencia en un bloque de referencia, y el bloque actual es codificado con base en las muestras de luminancia de referencia. En dicho caso, la predicción de vector de movimiento de luminancia temporal es ejecutada de acuerdo con:

$$xColBr = xCb + cbWidth;$$

```

yColBr = yCb + cbWidth;
rightBoundaryPos = subpic_treated_as_pic_flag[ Sub-PicIdx ] ?
SubPicRightBoundaryPos: pic_width_in_luma_samples — 1; e
botBoundaryPos = subpic_treated_as_pic_flag[ Sub-PicIdx ] ?
5 SubPicBotBoundaryPos: pic_height_in_luma_samples — 1,

```

donde xColBr e yColBr especifican una ubicación del bloque colocado, xCb e yCb especifican una muestra superior izquierda del bloque actual con relación a una muestra superior izquierda de la imagen actual, cbWidth es una anchura del bloque actual, cbHeight es una altura del bloque actual, SubPicRightBoundaryPos es una posición de un límite derecho de la sub-imagen, SubPicBotBoundaryPos es una posición de un límite inferior de la sub-imagen, pic_width_in_luma_samples es una anchura de la imagen actual medida en muestras de luminancia, pic_height_in_luma_samples es una altura de la imagen actual medida en muestras de luminancia, botBoundaryPos es una posición calculada del límite inferior de la sub-imagen, rightBoundaryPos es una posición calculada del límite derecho de la sub-imagen, Sub-PicIdx es un índice de la sub-imagen, y en donde vectores de movimiento colocados son excluidos cuando yCb >> CtbLog2SizeY no es igual a yColBr >> CtbLog2SizeY, donde CtbLog2SizeY indica un tamaño de un bloque de árbol de codificación .

La sub-imagen tratada como un indicador de imagen 1131 también puede ser empleada para una segunda implementación ejemplar. Al igual que en el primer ejemplo, la sub-imagen tratada como un indicador de imagen 1131 puede ser empleada cuando una sub-imagen actual en una imagen actual ha sido codificada de acuerdo con inter-predicción. En este ejemplo, un vector de movimiento puede ser determinado para un bloque actual de la sub-imagen (por ejemplo, de entre una lista de candidatos). Cuando se establece la sub-imagen tratada como un indicador de imagen 1131, se puede aplicar una función de recorte a ubicaciones de muestra en un bloque de referencia. Una ubicación de muestra es una posición en una imagen que puede contener una sola muestra incluyendo un valor de luminancia y/o un par de valores de crominancia. Un filtro de interpolación entonces puede ser aplicado cuando el vector de movimiento apunta fuera de la sub-imagen actual. Esta función de recorte garantiza que el filtro de interpolación no se base en datos de sub-imágenes adyacentes para mantener la separación entre las sub-imágenes para soportar extracción separada.

La función de recorte puede ser aplicada en un proceso de interpolación bilineal de muestra de luminancia. El proceso de interpolación bilineal de muestra de luminancia puede recibir entradas incluyendo una ubicación de luminancia en unidades de muestra completas (xIntL, yIntL). El proceso de interpolación bilineal de muestra de luminancia emite un valor de muestra de luminancia predicho (predSampleLXL). La función de recorte es aplicada a las ubicaciones de muestra de la siguiente manera. Cuando subpic_treated_as_pic_flag[Sub-PicIdx] es igual a uno, aplica lo siguiente:

```

xInti = Clip3( SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i ),

```

```

e
yInti = Clip3( SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i ),

```

donde subpic_treated_as_pic_flag es el indicador establecido para indicar que la sub-imagen es tratada como una imagen, Sub-PicIdx es un índice de la sub-imagen, xInti e yInti son una ubicación de muestra recortada en el índice i, SubPicRightBoundaryPos es una posición de un límite derecho de la sub-imagen, SubPicLeftBoundaryPos es una posición de un límite izquierdo de la sub-imagen, SubPicTopBoundaryPos es una posición de un límite superior de la sub-imagen, SubPicBotBoundaryPos es una posición de un límite inferior de la sub-imagen, y Clip3 es la función de recorte de acuerdo con:

$$\text{Clip3}(x, y, z) = \begin{cases} x & : z < x \\ y & : z > y \\ z & : \text{de otra manera} \end{cases}$$

donde x, y, z son valores de entrada numéricos.

La función de recorte también se puede aplicar en un proceso de filtración de interpolación de ocho derivaciones de muestra de luminancia. El proceso de filtración de interpolación de ocho derivaciones de muestra de luminancia recibe entradas incluyendo una ubicación de luminancia en unidades de muestra completas (xIntL, yIntL). El proceso de interpolación bilineal de muestra de luminancia emite un valor de muestra de luminancia predicho (predSampleLXL). La función de recorte es aplicada a las ubicaciones de muestra de la siguiente manera. Cuando subpic_treated_as_pic_flag[Sub-PicIdx] es igual a uno, aplica lo siguiente:

```

xInti = Clip3( SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i -3),

```

```

e
yInti = Clip3( SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i -3),

```

donde subpic_treated_as_pic_flag es el indicador establecido para indicar que la sub-imagen es tratada como una imagen, Sub-PicIdx es un índice de la sub-imagen, xInti e yInti son una ubicación de muestra recortada en el índice i, SubPicRightBoundaryPos es una posición de un límite derecho de la sub-imagen, SubPicLeftBoundaryPos es una posición de un límite izquierdo de la sub-imagen, SubPicTopBoundaryPos es una posición de un límite superior de la

sub-imagen, SubPicBotBoundaryPos es una posición de un límite inferior de la sub-imagen, y Clip3 es como se describió antes.

La función de recorte también se puede aplicar en un proceso de interpolación de muestra de crominancia. El proceso de interpolación de muestra de crominancia recibe entradas incluyendo una ubicación de crominancia en unidades de muestra completas (xIntC, yIntC). El proceso de interpolación de muestra de crominancia emite un valor de muestra de crominancia predicho (predMuestraLXC). La función de recorte es aplicada a las ubicaciones de muestra de la siguiente manera. Cuando subpic_treated_as_pic_flag[Sub-PicIdx] es igual a uno, aplica lo siguiente:

xInti = Clip3(SubPicLeftBoundaryPos/SubWidthC, SubPicRightBoundaryPos/SubWidthC, xIntC + i),

yInti = Clip3(SubPicTopBoundaryPos/SubHeightC, SubPicBotBoundaryPos/SubHeightC, yIntC + i),

donde subpic_treated_as_pic_flag es el indicador establecido para indicar que la sub-imagen es tratada como una imagen, Sub-PicIdx es un índice de la sub-imagen, xInti e yInti son una ubicación de muestra recortada en el índice i, SubPicRightBoundaryPos es una posición de un límite derecho de la sub-imagen, SubPicLeftBoundaryPos es una posición de un límite izquierdo de la sub-imagen, SubPicTopBoundaryPos es una posición de un límite superior de la sub-imagen, SubPicBotBoundaryPos es una posición de un límite inferior de la sub-imagen, SubWidthC y SubHeightC indican una relación de tasa de muestreo horizontal y vertical entre muestras de luminancia y crominancia, y Clip3 es como se describió antes.

Un indicador de filtro de bucle a través de sub-imagen habilitada 1132 en el SPS 1110 puede ser empleado para una tercera implementación ejemplar. El indicador de filtro de bucle a través de sub-imagen habilitada 1132 puede ser establecido para controlar si la filtración es empleada a través de límites de sub-imágenes especificadas. Por ejemplo, el indicador de filtro de bucle a través de sub-imagen habilitada 1132 puede ser denotado como un loop_filter_across_subpic_enabled_flag. El indicador de filtro de bucle a través de sub-imagen habilitada 1132 puede ser establecido a uno cuando se especifica que las operaciones de filtración en-bucle pueden ser ejecutadas a través de los límites de la sub-imagen o puesta a cero cuando se especifica que operaciones de filtración en-bucle no son ejecutadas a través de los límites de la sub-imagen. Por lo tanto, operaciones de filtración pueden o no ser ejecutadas a través del límite de sub-imagen con base en el valor del indicador de filtro de bucle a través de sub-imagen habilitada 1132. Las operaciones de filtración pueden incluir la aplicación de un filtro de desbloqueo 1043, un ALF 1047, y/o un filtro SAO 1045. De esta manera, los filtros pueden ser apagados para sub-imágenes que son codificadas para extracción separada o encendidas para sub-imágenes que son codificadas para despliegue como un grupo.

Un indicador de filtro de bucle a través de mosaicos habilitados 1134 en el PPS 1111 puede ser empleado para una cuarta implementación ejemplar. El indicador de filtro de bucle a través de mosaicos habilitados 1134 puede ser establecido para controlar si la filtración es empleada a través de los límites de mosaicos especificados. Por ejemplo, el indicador de filtro de bucle a través de mosaicos habilitados 1134 puede ser denotado como un loop_filter_across_tiles_enabled_flag. El indicador de filtro de bucle a través de mosaicos habilitados 1134 puede ser establecido a uno cuando se especifica que operaciones de filtración en-bucle pueden ser ejecutadas a través de los límites del mosaico o puesto a cero cuando se especifica que operaciones de filtración en-bucle no son ejecutadas a través de los límites del mosaico. Por lo tanto, operaciones de filtración pueden o no ser ejecutadas a través de los límites de mosaico especificados con base en el valor del indicador de filtro de bucle a través de mosaicos habilitados 1134. Las operaciones de filtración pueden incluir la aplicación de un filtro de desbloqueo 1043, un ALF 1047, y/o un filtro SAO 1045.

Los datos de sub-imagen 1133 en el SPS 1110 pueden ser empleados para una quinta implementación ejemplar. Los datos de sub-imagen 1133 pueden incluir una anchura, altura, y compensación para cada sub-imagen en los datos de imagen 1120. Por ejemplo, la anchura y altura de cada sub-imagen pueden ser descritas en los datos de sub-imagen 1133 en unidades de CTBs. En algunos ejemplos, la anchura y altura de la sub-imagen son almacenadas en los datos de sub-imagen 1133 as sub-imagen_anchura_menos1 y sub-imagen_altura_menos1, respectivamente. Además, una compensación de cada sub-imagen puede ser descrita en los datos de sub-imagen 1133 en unidades de CTUs. Por ejemplo, la compensación de cada sub-imagen puede ser especificada como una posición vertical y una posición horizontal de la CTU superior izquierda de la sub-imagen. Específicamente, la compensación de la sub-imagen puede ser especificada como una diferencia entre la CTU superior izquierda de la imagen y la CTU superior izquierda de la sub-imagen. En algunos ejemplos, la posición vertical y la posición horizontal de la CTU superior izquierda de la sub-imagen es almacenada en los datos de sub-imagen 1133 como una subpic_ctu_top_left_y y subpic_ctu_top_left_x, respectivamente. Esta implementación ejemplar describe las sub-imágenes en los datos de sub-imagen 1133 en términos de CTBs/CTUs en lugar de hacerlo en términos de mosaicos. Esto permite que sub-imágenes sean empleadas incluso cuando los mosaicos no son empleados en la imagen/sub-imagen correspondiente.

Los datos de sub-imagen 1133 en el SPS 1110, una dirección de segmento 1136 en la cabecera de segmento 1115, y un ID de sub-imagen de segmento 1135 en la cabecera de segmento 1115 pueden ser empleados para una sexta

implementación ejemplar. Los datos de sub-imagen 1133 pueden ser implementados como se describió en la quinta implementación ejemplar. La dirección de segmento 1136 puede incluir un índice de segmento de nivel de sub-imagen de un segmento (por ejemplo, en datos de imagen 1120) asociado con la cabecera de segmento 1115. Por ejemplo, el segmento es indexado con base en la posición del segmento en la sub-imagen en lugar de hacerlo con base en la posición de segmento en la imagen. La dirección de segmento 1136 puede ser almacenada en una variable `slice_address`. El ID de sub-imagen de segmento 1135 incluye un ID de la sub-imagen que contiene el segmento asociado con la cabecera de segmento 1115. Específicamente, el ID de sub-imagen de segmento 1135 puede hacer referencia a la descripción (por ejemplo, anchura, altura y compensación) de la sub-imagen correspondiente en los datos de sub-imagen 1133. El ID de sub-imagen de segmento 1135 puede ser almacenado en una variable `slice_subpic_id`. Por consiguiente, la dirección de segmento 1136 es señalizada como un índice con base en la posición del segmento en la sub-imagen denotada por el ID de sub-imagen de segmento 1135 como se describe en los datos de sub-imagen 1133. De esta manera, la posición del segmento en la sub-imagen puede ser determinada incluso cuando la sub-imagen es extraída separadamente y otras sub-imágenes son omitidas de la corriente de bits 1100. Esto se debe a que este esquema de direccionamiento segrega cada una de las direcciones de sub-imagen de otras sub-imágenes. Por consiguiente, la cabecera de segmento 1115 no necesita ser reescrita cuando la sub-imagen es extraída tal como se requeriría en un esquema de direccionamiento donde el segmento es direccionado con base en la posición del segmento en la imagen. Se debiera observar que este enfoque puede ser empleado cuando el segmento es un segmento rectangular/cuadrado (en oposición a un segmento de escaneo de trama). Por ejemplo, un `rect_slice_flag` en el PPS 1111 puede ser establecido igual a uno para indicar que el segmento es un segmento rectangular/cuadrado.

Una implementación ejemplar de sub-imágenes utilizadas en algunos sistemas de codificación de vídeo es de la siguiente manera. Información relacionada con sub-imágenes que pueden estar presentes en un CVS pueden ser señalizadas en un SPS. Dicha señalización puede incluir la siguiente información. El número de sub-imágenes que están presentes en cada imagen de los CVS puede estar incluido en el SPS. En el contexto del SPS o un CVS, las sub-imágenes colocadas para todas las unidades de acceso (AUs) colectivamente se pueden referir como una secuencia de sub-imagen. Un bucle para especificar adicionalmente información relacionada con propiedades de cada sub-imagen también se puede incluir en el SPS. Dicha información puede incluir la identificación de sub-imagen, la ubicación de sub-imagen (por ejemplo, la distancia de compensación entre la muestra de luminancia de esquina superior-izquierda de la sub-imagen y la muestra de luminancia de esquina superior-izquierda de la imagen), y el tamaño de la sub-imagen. Además, el SPS también puede ser empleado para señalar si cada una de las sub-imágenes es una sub-imagen de movimiento restringido, donde una sub-imagen de movimiento restringido es una sub-imagen que contiene un MCTS. Información de perfil, tramo y nivel para cada sub-imagen se puede incluir en una corriente de bits a menos que dicha información se pueda derivar de otra manera. Dicha información puede ser empleada para información de perfil, tramo y nivel para una corriente de bits extraída creada a partir de la extracción de la sub-imagen de la corriente de bits original incluyendo la imagen completa. El perfil y tramo de cada sub-imagen pueden ser derivados para que sean los mismos que el perfil y tramo del original. El nivel para cada sub-imagen puede ser señalizado de manera explícita. Dicha señalización puede estar presente en el bucle antes descrito. Parámetros de decodificador de referencia hipotético (HRD) a nivel de secuencia, pueden ser señalizados en una porción de información de utilidad de vídeo (VUI) del SPS para cada sub-imagen (o de manera equivalente, cada secuencia de sub-imagen).

Cuando una imagen no es dividida en dos o más sub-imágenes, las propiedades de la sub-imagen (por ejemplo, ubicación, tamaño, etc.), excepto el ID de sub-imagen, no pueden ser señalizadas en la corriente de bits. Cuando se extrae una sub-imagen en imágenes en un CVS, cada unidad de acceso en la nueva corriente de bits puede no contener sub-imágenes, debido a que los datos de imagen resultantes en cada AU en la nueva corriente de bits no son divididos en múltiples sub-imágenes. Por lo tanto, las propiedades de sub-imagen tal como la ubicación y el tamaño se pueden omitir del SPS debido a que dicha información puede ser derivada de las propiedades de imagen. Sin embargo, la identificación de sub-imagen sigue siendo señalizada debido a que este ID puede ser referido por unidades de capa de codificación de vídeo (VCL) NAL / grupos de mosaicos incluidos en la sub-imagen extraída. El cambio del ID de sub-imagen debiera evitarse cuando se extrae la sub-imagen para reducir el uso de recursos.

La ubicación de una sub-imagen en la imagen (compensación x y compensación y) puede ser señalizada en unidades de muestras de luminancia y puede representar la distancia entre la muestra de luminancia de esquina superior-izquierda de la sub-imagen y la muestra de luminancia de esquina superior-izquierda de la imagen. En otro ejemplo, la ubicación de una sub-imagen en la imagen puede ser señalizada en unidades del tamaño de bloque de luminancia de codificación mínimo (`MinCbSizeY`) y puede representar la distancia entre la muestra de luminancia de esquina superior-izquierda de la sub-imagen y la muestra de luminancia de esquina superior-izquierda de la imagen. En otro ejemplo, la unidad de compensaciones de ubicación de sub-imagen puede ser explícitamente indicada por un elemento de sintaxis en un conjunto de parámetros, y la unidad puede ser `CtbSizeY`, `MinCbSizeY`, muestra de luminancia u otros valores. El códec puede requerir que cuando un margen derecho de sub-imagen no coincide con el margen derecho de imagen, la anchura de sub-imagen deberá ser un entero múltiplo del tamaño de CTU de luminancia (`CtbSizeY`). De igual manera, el códec además puede requerir que cuando un margen inferior de sub-

imagen no coincide con el margen inferior de imagen, la altura de sub-imagen debiera ser un entero múltiplo de CtbSizeY. El códec también puede requerir que cuando una anchura de sub-imagen no es un entero múltiplo de tamaño de CTU de luminancia, la sub-imagen está ubicada en la posición más a la derecha en la imagen. De igual manera, el códec también puede requerir que la sub-imagen esté localizada en la posición más al fondo en la imagen cuando la altura de sub-imagen no es un entero múltiplo de tamaño de CTU de luminancia. Cuando una anchura de sub-imagen es señalizada en unidades de tamaño de CTU de luminancia y la anchura de la sub-imagen no es un entero múltiplo de tamaño de CTU de luminancia, la anchura real en muestras de luminancia puede ser derivada con base en la ubicación de compensación de sub-imagen, la anchura de sub-imagen en tamaño de CTU de luminancia, y la anchura de imagen en muestras de luminancia. De igual manera, cuando la altura de sub-imagen es señalizada en unidades de tamaño de CTU de luminancia y la altura de la sub-imagen no es un entero múltiplo de tamaño de CTU de luminancia, la altura real en muestras de luminancia puede ser derivada con base en la ubicación de compensación de sub-imagen, la altura de sub-imagen en tamaño de CTU de luminancia, y la altura de imagen en muestras de luminancia.

Para cualquier sub-imagen, el ID de sub-imagen puede ser diferente del índice de sub-imagen. El índice de sub-imagen puede ser el índice de la sub-imagen tal como es señalizado en el bucle de sub-imágenes en el SPS. Alternativamente, el índice de sub-imagen puede ser un índice asignado en el orden de escaneo de trama de sub-imagen con relación a la imagen. Cuando el valor del ID de sub-imagen de cada sub-imagen es el mismo que su índice de sub-imagen, el ID de sub-imagen puede ser señalizado o derivado. Cuando el ID de sub-imagen de cada sub-imagen es diferente de su índice de sub-imagen, el ID de sub-imagen es explícitamente señalizado. El número de bits para señalar los IDs de sub-imagen puede ser señalizado en el mismo conjunto de parámetros que contiene propiedades de sub-imagen (por ejemplo, en el SPS). Algunos valores para ID de sub-imagen pueden ser reservados para algunos propósitos. Dicha reservación de valor puede ser de la siguiente manera. Cuando el grupo de mosaicos/cabeceras de segmento contienen un ID de sub-imagen para especificar cuál sub-imagen incluye el grupo de mosaicos, el valor cero puede ser reservado y no puede ser utilizado para sub-imágenes para garantizar que los primeros pocos bits al inicio de un grupo de mosaicos/cabecera de segmento no sean todos ceros con el fin de evitar la generación de un código de prevención de emulación. Cuando las sub-imágenes de una imagen no cubren toda el área de la imagen sin traslape e intervalo, se puede reservar un valor (por ejemplo, valor uno) para grupos de mosaicos que no son parte de ninguna de las sub-imágenes. Alternativamente, el ID de sub-imagen del área remanente puede ser explícitamente señalizada. El número de bits para señalar el ID de sub-imagen puede ser restringido de la siguiente manera. El rango de valor debiera ser suficiente para identificar de manera única todas las sub-imágenes en una imagen, incluyendo los valores reservados de ID de sub-imagen. Por ejemplo, el número mínimo de bits para un ID de sub-imagen puede ser el valor de Techo($\log_2(\text{número de sub-imágenes en una imagen} + \text{número de ID de sub-imagen reservado})$).

La unión de sub-imágenes en un bucle puede ser requerida para cubrir toda la imagen sin intervalo y sin traslape. Cuando esta restricción es aplicada, un indicador está presente para que cada sub-imagen especifique si la sub-imagen es una sub-imagen de movimiento restringido lo cual significa que la sub-imagen puede ser extraída. Alternativamente, la unión de sub-imágenes no puede cubrir toda la imagen. Sin embargo, puede no haber traslape entre sub-imágenes de una imagen.

Los IDs de sub-imagen pueden estar presentes inmediatamente después de una cabecera de unidad NAL para ayudar en el proceso de extracción de sub-imagen de manera que el extractor no necesita entender el resto de los bits de la unidad NAL. Para unidades NAL VCL, el ID de sub-imagen puede estar presente en los primeros bits de cabeceras de grupos de mosaicos. Para unidades NAL no-VCL, puede aplicar lo siguiente. Puede no requerirse que el ID de sub-imagen esté presente inmediatamente después de la cabecera de unidad NAL para el SPS. Con relación al PPS, cuando todos los grupos de mosaicos de la misma imagen están restringidos para hacer referencia al mismo PPS, no hay necesidad de que el ID de sub-imagen esté presente inmediatamente después de la cabecera de unidad NAL. Por otra parte, si los grupos de mosaicos de la misma imagen tienen permitido hacer referencia a diferentes PPSs, el ID de sub-imagen puede estar presente en los primeros bits de PPS (por ejemplo, inmediatamente después de la cabecera de unidad NAL PPS). En este caso, dos grupos de mosaicos diferentes de una imagen no tienen permitido compartir el mismo PPS. Alternativamente, cuando grupos de mosaicos de la misma imagen tienen permitido hacer referencia a diferentes PPSs, y diferentes grupos de mosaicos de la misma imagen también tienen permitido compartir el mismo PPS, ningún ID de sub-imagen está presente en la sintaxis PPS. Alternativamente, cuando grupos de mosaicos de la misma imagen tienen permitido hacer referencia a diferentes PPSs y diferentes grupos de mosaicos de la misma imagen también tienen permitido compartir el mismo PPS, una lista de IDs de sub-imagen está presente en la sintaxis PPS. La lista indica las sub-imágenes a las cuales aplica el PPS. Para otras unidades NAL no-VCL, si la unidad no-VCL aplica al nivel de imagen (por ejemplo, delimitador de unidad de acceso, fin de secuencia, fin de corriente de bits, etc.) o arriba, entonces no hay necesidad de que un ID de sub-imagen esté presente inmediatamente después de su cabecera de unidad NAL. De otra manera, el ID de sub-imagen puede estar presente inmediatamente después de la cabecera de unidad NAL.

La división de mosaicos dentro de sub-imágenes individuales puede ser señalizada en el PPS, pero grupos de mosaicos dentro de la misma imagen tienen permitido hacer referencia a diferentes PPSs. En este caso, los mosaicos son agrupados dentro de cada sub-imagen en lugar de hacer a través de la imagen. Por consiguiente, el concepto de agrupamiento de mosaicos en dicho caso incluye una división de una sub-imagen en mosaicos. Alternativamente, se puede emplear un Conjunto de Parámetros de Sub-Imagen (SPPS) para describir la división de mosaicos dentro de sub-imágenes individuales. Un SPPS se refiere a un SPS empleando un elemento de sintaxis que se refiere al ID de SPS. Un SPPS puede contener ID de sub-imagen. Para propósitos de extracción de sub-imagen, el elemento de sintaxis que se refiere al ID de sub-imagen es el primer elemento de sintaxis en SPPS. El SPPS contiene una estructura de mosaico que indica un número de columnas, un número de filas, separación de mosaicos uniforme, etc. El SPPS puede contener un indicador para indicar si el filtro de bucle está o no habilitado a través de los límites de sub-imagen asociados. De manera alternativa, las propiedades de sub-imagen para cada sub-imagen pueden ser señalizadas en el SPPS en lugar de hacerlo en el SPS. La división de mosaicos dentro de sub-imágenes individuales puede ser señalizada en el PPS, pero grupos de mosaicos dentro de la misma imagen tienen permitido hacer referencia a diferentes PPSs. Una vez activado, un SPPS puede durar por una secuencia de AUs consecutivas en orden de decodificación, pero puede ser desactivado/activado en una AU que no es el inicio de un CVS. Múltiples SPPSs pueden estar activas en cualquier momento durante el proceso de decodificación de una corriente de bits de capa única con múltiples sub-imágenes, y un SPPS puede ser compartido por diferentes sub-imágenes de una AU. Alternativamente, el SPPS y el PPS se pueden fusionar en un conjunto de parámetros. Para que esto ocurra, todos los grupos de mosaicos que están incluidos en la misma sub-imagen pueden estar restringidos a hacer referencia al mismo conjunto de parámetros que resulta de la fusión entre SPPS y PPS.

El número de bits utilizados para señalar el ID de sub-imagen puede ser señalizado en la cabecera de unidad NAL. Dicha información, cuando está presente, ayuda en el proceso de extracción de sub-imagen en el análisis de valores de ID de sub-imagen para el inicio de una carga útil de unidad NAL (por ejemplo, los primeros pocos bits inmediatamente después de la cabecera de unidad NAL). Para dicha señalización, algunos de los bits reservados en una cabecera de unidad NAL pueden ser utilizados para evitar el incremento de la longitud de la cabecera de unidad NAL. El número de bits para dicha señalización debiera cubrir el valor de sub-imagen-ID-bit-len. Por ejemplo, cuatro bits de entre siete bits reservados en la cabecera de unidad NAL de los VVCs pueden ser utilizados para este propósito.

Cuando se decodifica una sub-imagen, la ubicación de cada bloque de árbol de codificación, denotada como posición CTB vertical (xCtb) y posición CTB horizontal (yCtb), son ajustadas a una ubicación de muestra de luminancia real en la imagen en lugar de una ubicación de muestra de luminancia en la sub-imagen. De esta manera, la extracción de la sub-imagen colocada de cada imagen de referencia se puede evitar a medida que todo es decodificado como si estuviera localizado en la imagen en lugar de estar en la sub-imagen. Para ajustar la ubicación del bloque de árbol de codificación, las variables SubpictureXOffset y SubpictureYOffset son derivadas con base en la posición de sub-imagen (subpic_x_offset y subpic_y_offset). Los valores de las variables son agregados a los valores de las coordenadas x e y de ubicación de muestra de luminancia, respectivamente, de cada bloque de árbol de codificación en la sub-imagen. El proceso de extracción de sub-imagen puede ser definido de la siguiente manera. La entrada al proceso incluye la sub-imagen objetivo que se va a extraer. Esto puede ser introducido en la forma de un ID de sub-imagen o una ubicación de sub-imagen. Cuando la entrada es la ubicación de sub-imagen, el ID de sub-imagen asociado puede ser resuelto analizando la información de sub-imagen en el SPS. Para unidades NAL no-VCL, aplica lo siguiente. Los elementos de sintaxis en el SPS relacionado con el tamaño y el nivel de imagen son actualizados con la información de tamaño y nivel de sub-imagen. Las siguientes unidades NAL no-VCL no son alteradas por extracción: PPS, delimitador de unidad de acceso (AUD), fin de secuencia (EOS), fin de corriente de bits (EOB), y cualesquiera otras unidades NAL no-VCL que son aplicables a nivel de imagen o por arriba. Las unidades NAL no-VCL remanentes con ID de sub-imagen no igual al ID de sub-imagen objetivo son eliminadas. Unidades NAL VCL con ID de sub-imagen no igual al ID de sub-imagen objetivo también son eliminadas.

Un mensaje SEI de anidado de sub-imagen puede ser utilizado para anidar mensajes SEI a nivel de AU o a nivel de sub-imagen para un conjunto de sub-imágenes. Los datos llevados en el mensaje SEI de anidado de sub-imagen pueden incluir el periodo de almacenamiento en memoria tampón, temporización de imagen, y mensajes SEI no-HRD. La sintaxis y semántica de este mensaje SEI pueden ser de la siguiente manera. Para operaciones de sistemas, tal como ambientes de formato de medios omnidireccionales (OMAF), un conjunto de secuencias de sub-imagen que cubre una ventanilla de observación puede ser solicitado y decodificado por el reproductor OMAF. Por lo tanto, un mensaje SEI a nivel de secuencia puede llevar información de un conjunto de secuencias de sub-imagen que colectivamente incluyen una región de imagen rectangular o cuadrada. La información puede ser utilizada por sistemas, y la información es indicativa de la capacidad de decodificación mínima así como la tasa de bits del conjunto de secuencias de sub-imagen. La información incluye el nivel de la corriente de bits incluyendo solamente el conjunto de secuencias de sub-imagen, la tasa de bits de la corriente de bits, y opcionalmente un proceso de extracción de sub-corriente de bits especificado para el conjunto de secuencias de sub-imagen.

La implementación precedente incluye varios problemas. La señalización de la anchura y altura de la imagen y/o las anchuras/alturas/compensaciones de las sub-imágenes no es eficiente. Se pueden ahorrar más bits para señalar dicha información. Cuando la información de tamaño y posición de sub-imagen es señalizada en el SPS, el PPS incluye la configuración de mosaico. Además, un PPS tiene permitido ser compartido por múltiples sub-imágenes de la misma imagen. Por lo tanto, los rangos de valor para `num_tile_columns_minus1` y `num_tile_rows_minus1` debieran ser especificados de manera más clara. Además, la semántica del indicador que indica si una sub-imagen tiene movimiento restringido no se especifica claramente. El nivel es señalizado de manera obligatoria para cada secuencia de sub-imagen. Sin embargo, cuando una secuencia de sub-imagen no puede ser decodificada de manera independiente, la señalización del nivel de la sub-imagen no es útil. Además, en algunas aplicaciones algunas secuencias de sub-imagen debieran ser decodificadas y entregadas junto con al menos otra secuencia de sub-imagen. Por lo tanto, la señalización de un nivel para una sola de dichas secuencias de sub-imagen puede no ser útil. Además, la determinación de un valor de nivel para cada sub-imagen puede sobrecargar el codificador.

Con la introducción de secuencias de sub-imagen independientemente decodificables, escenarios que requieren la extracción y decodificación independiente de algunas regiones de las imágenes no pueden funcionar con base en los grupos de mosaicos. Por lo tanto, la señalización explícita de IDs de grupos de mosaico puede no ser útil. Además, el valor de cada uno de los elementos de sintaxis PPS `pps_seq_parameter_id` y `loop_filter_across_tiles_enabled_flag` debiera ser el mismo en todos los PPSs a los que hacen referencia las cabeceras de grupos de mosaicos de una imagen codificada. Esto se debe a que el SPS active no debiera cambiar dentro de un CVS, y el valor de `loop_filter_across_tiles_enabled_flag` debiera ser el mismo para todos los mosaicos dentro de una imagen para procesamiento paralelo con base en mosaicos. Si se permite el mezclado de grupos de mosaicos rectangulares y de escaneo de trama dentro de una imagen, esto debiera ser claramente especificado. También se debiera especificar si se permiten sub-imágenes que sean parte de diferentes imágenes y empleen el mismo ID de sub-imagen en un CVS para utilizar diferentes modos de grupos de mosaicos. El proceso de derivación para predicción de vector de movimiento de luminancia temporal puede no permitir el trato de los límites de sub-imagen como límites de imagen en predicción de vector de movimiento temporal (TMVP). Además, el proceso de interpolación bilineal de muestra de luminancia, el proceso de filtración de interpolación de 8 derivaciones de muestra de luminancia, y el proceso de interpolación de muestra de crominancia no se pueden configurar para tratar los límites de sub-imagen como límites de imagen en compensación de movimiento. También se debiera especificar un mecanismo para control de operaciones de filtración de desbloqueo, SAO, y ALF en los límites de sub-imagen.

Con la introducción de secuencias de sub-imagen independientemente decodificables, el `loop_filter_across_tile_groups_enabled_flag` puede ser menos útil. Esto se debe a que el apagado de las operaciones de filtración en-bucle para propósitos de procesamiento paralelo también se puede satisfacer estableciendo `loop_filter_across_tile_groups_enabled_flag` igual a cero. Además, el apagado de las operaciones de filtración en-bucle para habilitar la extracción independiente y la decodificación de algunas regiones de las imágenes también se puede satisfacer estableciendo `loop_filter_across_sub_pic_enabled_flag` igual a cero. Por lo tanto, la especificación adicional de los procesos para el apagado de las operaciones de filtración en-bucle a través de los límites de grupo de mosaicos con base en `loop_filter_across_tile_groups_enabled_flag` innecesariamente cargaría el decodificador y desperdiciaría bits. Además, el proceso de decodificación conforme a lo antes especificado no puede permitir el apagado de las operaciones de filtración ALF a través de los límites de mosaico.

Debido a esto, la presente divulgación incluye diseños para soportar codificación de vídeo basada en sub-imágenes. Una sub-imagen es un área rectangular o cuadrada dentro de una imagen que puede o no ser decodificada independientemente utilizando el mismo proceso de decodificación que una imagen. Las descripciones de las técnicas están basadas en la norma de Codificación de Vídeo Versátil (VC). Sin embargo, las técnicas también pueden aplicar a otras especificaciones de códec de vídeo.

En algunos ejemplos, una unidad de tamaño es señalizada para los elementos de sintaxis de anchura y altura de imagen y la lista de elementos de anchura/altura/compensación_x/compensación_y de sub-imagen. Todos los elementos de sintaxis son señalizados en la forma de `xxx_menos 1`. Por ejemplo, cuando la unidad de tamaño es sesenta y cuatro muestras de luminancia, un valor de anchura de noventa y nueve especifica una anchura de imagen de seis mil cuatrocientos muestras de luminancia. El mismo ejemplo aplica para otros de estos elementos de sintaxis. En otro ejemplo, aplica uno o más de los siguientes. Una unidad de tamaño puede ser señalizada para los elementos de sintaxis de anchura y altura de imagen en la forma de `xxx_menos 1`. Dicha unidad de tamaño señalizada para la lista de elementos de anchura/altura/compensación_x/compensación_y de sub-imagen puede ser en la forma de `xxx_menos 1`. En otro ejemplo, aplica uno o más de los siguientes. Una unidad de tamaño para la anchura de imagen y la lista de elementos de sintaxis de anchura/compensación_x de sub-imagen puede ser señalizada en la forma de `xxx_menos 1`. Una unidad de tamaño para la altura de imagen y la lista de elementos de sintaxis de altura/compensación_y de sub-imagen puede ser señalizada en la forma de `xxx_menos 1`. En otro ejemplo, aplica uno o más de los siguientes. Los elementos de sintaxis de anchura y altura de imagen en la forma de `xxx_menos1` pueden ser señalizados en unidades de codificación mínimas. Los elementos de sintaxis de anchura/altura/compensación_x/compensación_y de sub-imagen en la forma de `xxx_menos 1` pueden ser

señalizados en unidades de CTUs o CTBs. Se puede derivar la anchura de sub-imagen para cada sub-imagen en el límite de imagen derecha. La altura de sub-imagen para cada sub-imagen en el límite de imagen inferior puede ser derivada. Otros valores de la anchura/altura/compensación_x/compensación_y de sub-imagen pueden ser señalizados en la corriente de bits. En otros ejemplos, un modo para señalización de las anchuras y alturas de las sub-imágenes así como sus posiciones dentro de la imagen pueden ser agregados para casos cuando las sub-imágenes tienen un tamaño uniforme. Las sub-imágenes tienen un tamaño uniforme cuando incluyen las mismas filas de sub-imagen y las columnas de sub-imagen. En este modo, el número de filas de sub-imagen, el número de columnas de sub-imagen, la anchura de cada columna de sub-imagen, y la altura de cada fila de sub-imagen pueden ser señalizadas.

En otro ejemplo, la señalización de la anchura y altura de sub-imagen no puede ser incluida en el PPS. El `num_tile_columns_minus1` y `num_tile_rows_minus1` debieran estar en el rango de cero a uno de un valor entero tal como un mil veinticuatro, inclusivo. En otro ejemplo, cuando sub-imágenes que se refieren al PPS tienen más de un mosaico, dos elementos de sintaxis acondicionados por un indicador de presencia pueden ser señalizados en el PPS. Estos elementos de sintaxis son empleados para señalar la anchura y altura de sub-imagen en unidades de CTBs y especifican el tamaño de todas las sub-imágenes refiriéndose al PPS.

En otro ejemplo, también se puede señalar información adicional que describe sub-imágenes individuales. Un indicador, tal como un `sub_pic_treated_as_pic_flag[i]`, puede ser señalado para cada secuencia de sub-imagen para indicar si las sub-imágenes de la secuencia de sub-imagen son tratadas como imágenes en el proceso de decodificación para propósitos diferentes a las operaciones de filtración en-bucle. El nivel al cual una secuencia de sub-imagen de adapta puede ser únicamente señalado cuando el `sub_pic_treated_as_pic_flag[i]` es igual a uno. Una secuencia de sub-imagen es un CVS de sub-imágenes con el mismo ID de sub-imagen. Cuando el `sub_pic_treated_as_pic_flag[i]` es igual a uno, el nivel de la secuencia de sub-imagen también puede ser señalado. Esto puede ser controlado por un indicador para todas las secuencias de sub-imagen o por un indicador por cada secuencia de sub-imagen. En otro ejemplo, la extracción de sub-corriente de bits puede ser habilitada sin cambiar las unidades NAL VCL. Esto se puede lograr removiendo la señalización de ID de grupo de mosaicos explícito del PPS. Las semánticas de `tile_group_address` son especificadas cuando `rect_tile_group_flag` es igual a uno indicando grupos de mosaicos rectangulares. El `tile_group_address` puede incluir el índice de grupo de mosaicos del grupo de mosaicos entre grupos de mosaicos dentro de la sub-imagen.

En otro ejemplo, el valor de cada uno de los elementos de sintaxis `PPS pps_seq_parameter_id` y `loop_filter_across_tiles_enabled_flag` debiera ser el mismo en todos los PPSs referidos por las cabeceras de grupos de mosaicos de una imagen codificada. Otros elementos de sintaxis PPS pueden ser diferentes para diferentes PPSs referidos a las cabeceras de grupos de mosaicos de una imagen codificada. El valor de `single_tile_in_pic_flag` puede ser diferente para diferentes PPSs referidos a las cabeceras de grupos de mosaicos de una imagen codificada. De esta manera, algunas imágenes en un CVS pueden tener solamente un mosaico, mientras que algunas otras imágenes en un CVS pueden tener múltiples mosaicos. Esto también permite que algunas sub-imágenes de una imagen (por ejemplo, aquellas que son muy grandes) tengan múltiples mosaicos mientras que otras sub-imágenes de la misma imagen (por ejemplo, aquellas que son muy pequeñas) tienen solamente un mosaico.

En otro ejemplo, una imagen puede incluir una mezcla de grupos de mosaicos rectangulares/cuadrados y de escaneo de trama. Por consiguiente, algunas sub-imágenes de la imagen utilizan el modo de grupo de mosaicos rectangular/cuadrado, mientras que otras sub-imágenes utilizan el modo de grupo de mosaicos de escaneo de trama. Esta flexibilidad es beneficiosa para escenarios de fusión de corriente de bits. Alternativamente, una restricción puede requerir que todas las sub-imágenes de una imagen deban utilizar el mismo modo de grupo de mosaicos. Las sub-imágenes de diferentes imágenes con el mismo ID de sub-imagen en un CVS no pueden utilizar diferentes modos de grupo de mosaicos. Las sub-imágenes de diferentes imágenes con el mismo ID de sub-imagen en un CVS pueden utilizar diferentes modos de grupo de mosaicos.

En otro ejemplo, cuando la `sub_pic_treated_as_pic_flag[i]` para una sub-imagen es igual a uno, vectores de movimiento colocados para predicción de vector de movimiento temporal para la sub-imagen están limitados a provenir del interior de los límites de la sub-imagen. Por consiguiente, la predicción de vector de movimiento temporal para la sub-imagen es tratada como si los límites de sub-imagen fuesen límites de imagen. Además, operaciones de recorte son especificadas como parte del proceso de interpolación bilineal de muestra de luminancia, el proceso de filtración de interpolación de 8 derivaciones de muestra de luminancia, y el proceso de interpolación de muestra de crominancia, para permitir el trato de los límites de sub-imagen como límites de imagen en compensación de movimiento para sub-imágenes para las cuales el `sub_pic_treated_as_pic_flag[i]` es igual a uno.

En otro ejemplo, cada sub-imagen está asociada con un indicador señalado, tal como un `loop_filter_across_sub_pic_enabled_flag`. El indicador es empleado para control de operaciones de filtración en-bucle en los límites de la sub-imagen y para control de las operaciones de filtración en los procesos de

decodificación correspondientes. El proceso de filtro de desbloqueo puede no ser aplicado a bordes de sub-bloque de codificación y bordes de bloque de transformada que coinciden con los límites de sub-imágenes para los cuales `loop_filter_across_sub_pic_enabled_flag` es igual a cero. Alternativamente, el proceso de filtro de desbloqueo no es aplicado a bordes de sub-bloque de codificación y bordes de bloque de transformada que coinciden con los límites superior o izquierdo de sub-imágenes para los cuales `loop_filter_across_sub_pic_enabled_flag` es igual a cero. Alternativamente, el proceso de filtro de desbloqueo no es aplicado a bordes de sub-bloque de codificación y bordes de bloque de transformada que coinciden con los límites de sub-imágenes para los cuales `sub_pic_treated_as_pic_flag[i]` es igual a uno o cero. Alternativamente, el proceso de filtro de desbloqueo no es aplicado a bordes de sub-bloque de codificación y bordes de bloque de transformada que coinciden con los límites superior o izquierdo de sub-imágenes. Una operación de recorte puede ser especificada para apagar la operación de filtración SAO a través de los límites de una sub-imagen cuando `loop_filter_across_sub_pic_enabled_flag` para la sub-imagen es igual a cero. Una operación de recorte puede ser especificada para apagar la operación de filtración ALF a través de los límites de la sub-imagen cuando `loop_filter_across_sub_pic_enabled_flag` es igual a cero para una sub-imagen. El `loop_filter_across_tile_groups_enabled_flag` puede ser removido del PPS. Por consiguiente, cuando el `loop_filter_across_tiles_enabled_flag` es igual a cero, operaciones de filtración en-bucle a través de los límites de grupo de mosaicos que no son los límites de sub-imagen no son apagados. Las operaciones de filtro de bucle pueden incluir desbloqueo, SAO, y ALF. En otro ejemplo, una operación de recorte es especificada para apagar la operación de filtración ALF a través de los límites de un mosaico cuando `loop_filter_across_tiles_enabled_flag` para el mosaico es igual a cero.

Uno o más de los ejemplos precedentes pueden ser implementados de la siguiente manera. Una sub-imagen puede ser definida como una región rectangular/cuadrada de uno o más grupos de mosaicos o segmentos dentro de una imagen. Las siguientes divisiones de elementos de procesamiento pueden formar división espacial o a nivel de componente: la división de cada imagen en componentes, la división de cada componente en CTBs, la división de cada imagen en sub-imágenes, la división de cada sub-imagen en columnas de mosaico dentro de una sub-imagen, la división de cada sub-imagen en filas de mosaico dentro de una sub-imagen, la división de cada columna de mosaicos dentro de una sub-imagen en mosaicos, la división de cada fila de mosaicos dentro de una sub-imagen en mosaicos, y la división de cada sub-imagen en grupos de mosaicos.

El proceso para escaneo de mosaicos y rastreo CTB dentro de una sub-imagen puede ser de la siguiente manera. La lista `ColWidth[i]` para i que varía de cero a `num_tile_columns_minus1`, inclusivo, especificando la anchura de la i -ava columna de mosaicos en unidades de CTBs, puede ser derivada de la siguiente manera:

```
if( uniform_tile_spacing_flag )
for( i = 0; i <= num_tile_columns_minus1; i++ )
    ColWidth[ i ] = ( ( i + 1 ) * SubPicWidthInCtbsY ) / ( num_tile_columns_minus1 + 1 ) -
                    ( i * SubPicWidthInCtbsY ) / ( num_tile_columns_minus1 + 1 )
else {
ColWidth[ num_tile_columns_minus1 ] = SubPicWidthInCtbsY
for( i = 0; i < num_tile_columns_minus1; i++ ) {
    ColWidth[ i ] = tile_column_width_minus1[ i ] + 1
    ColWidth[ num_tile_columns_minus1 ] -= ColWidth[ i ]
}
}
```

La lista `RowHeight[j]` para j que varía de cero a `num_tile_rows_minus1`, inclusivo, especificando la altura de la j -ava fila de mosaicos en unidades de CTBs, es derivada de la siguiente manera:

```
if( uniform_tile_spacing_flag )
for( j = 0; j <= num_tile_rows_minus1; j++ )
    RowHeight[ j ] = ( ( j + 1 ) * SubPicHeightInCtbsY ) / ( num_tile_rows_minus1 + 1 ) -
                    ( j * SubPicHeightInCtbsY ) / ( num_tile_rows_minus1 + 1 )
else {
RowHeight[ num_tile_rows_minus1 ] = SubPicHeightInCtbsY
for( j = 0; j < num_tile_rows_minus1; j++ ) {
```

```

    RowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    RowHeight[ num_tile_rows_minus1 ] -= RowHeight[ j ]
}
}

```

La lista ColBd[i] para i variando de cero a num_tile_columns_minus1 + 1, inclusivo, especificando la ubicación del i-avo límite de columna de mosaico en unidades de CTBs, es derivado de la siguiente manera:

```

5   for( ColBd[ 0 ] = 0, i = 0; i <= num_tile_columns_minus1; i++ )
        ColBd[ i + 1 ] = ColBd[ i ] + ColWidth[ i ] (6-3)

```

La lista RowBd[j] para j variando de cero a num_tile_rows_minus1 + 1, inclusivo, especificando la ubicación del j-avo límite de fila de mosaicos en unidades de CTBs, es derivada de la siguiente manera:

```

    for( RowBd[ 0 ] = 0, j = 0; j <= num_tile_rows_minus1; j++ )
10   RowBd[ j + 1 ] = RowBd[ j ] + RowHeight[ j ] (6-4)

```

La lista CtbAddrRsATs[ctbAddrRs] para ctbAddrRs que varía de cero a SubPicSizInCtbsY – 1, inclusivo, especificando la conversión de una dirección CTB en escaneo de trama CTB de una sub-imagen a una dirección CTB en escaneo de mosaico de la sub-imagen, es derivado de la siguiente manera:

```

    for( ctbAddrRs = 0; ctbAddrRs < SubPicSizInCtbsY; ctbAddrRs++ ) {
        tbX = ctbAddrRs % SubPicWidthInCtbsY
        tbY = ctbAddrRs / SubPicWidthInCtbsY
        for( i = 0; i <= num_tile_columns_minus1; i++ )
            if( tbX >= ColBd[ i ] )
                tileX = i
        for( j = 0; j <= num_tile_rows_minus1; j++ ) (6-5)
            if( tbY >= RowBd[ j ] )
                tileY = j
        CtbAddrRsToTs[ ctbAddrRs ] = 0
        for( i = 0; i < tileX; i++ )
            CtbAddrRsToTs[ ctbAddrRs ] += RowHeight[ tileY ] * ColWidth[ i ]
        for( j = 0; j < tileY; j++ )
            CtbAddrRsToTs[ ctbAddrRs ] += SubPicWidthInCtbsY * RowHeight[ j ]
        CtbAddrRsToTs[ ctbAddrRs ] += ( tbY - RowBd[ tileY ] ) * ColWidth[ tileX ] + tbX - ColBd[ tileX ]
15   }

```

La lista CtbAddrTsToRs[ctbAddrTs] para ctbAddrTs que varía de cero a SubPicSizInCtbsY – 1, inclusivo, especificando la conversión de una dirección CTB en el escaneo de mosaico a una dirección CTB en el escaneo de trama CTB de una sub-imagen, es derivada de la siguiente manera:

```

    for( ctbAddrRs = 0; ctbAddrRs < SubPicSizInCtbsY; ctbAddrRs++ ) (6-6)
20   CtbAddrTsToRs[ CtbAddrRsToTs[ ctbAddrRs ] ] = ctbAddrRs

```

La lista TileIdx[ctbAddrTs] para ctbAddrTs que varía de 0 a SubPicSizInCtbsY – 1, inclusivo, especificando la conversión de una dirección CTB en el escaneo de mosaico de una sub-imagen a un ID de mosaico, es derivada de la siguiente manera:

```

    for( j = 0, tileIdx = 0; j <= num_tile_rows_minus1; j++ )
        for( i = 0; i <= num_tile_columns_minus1; i++, tileIdx++ )
25   for( y = RowBd[ j ]; y < RowBd[ j + 1 ]; y++ )(6-7)
        for( x = ColBd[ i ]; x < ColBd[ i + 1 ]; x++ )
            TileIdx[ CtbAddrRsToTs[ y * SubPicWidthInCtbsY + x ] ] = tileIdx

```

La lista NumCtusInTile[TileIdx] para TileIdx que varía de 0 a NumTilesInSubPic – 1, inclusivo, especificando la conversión de un índice de mosaico al número de CTUs en el mosaico, es derivado de la siguiente manera:

```

for( j = 0, tileIdx = 0; j <= num_tile_rows_minus1; j++ )
for( i = 0; i <= num_tile_columns_minus1; i++, tileIdx++ )
    NumCtusInTile[ tileIdx ] = ColWidth[ i ] * RowHeight[ j ]

```

(6-8)

La lista FirstCtbAddrTs[TileIdx] para TileIdx que varía de cero a NumTilesInSubPic – 1, inclusivo, especificando la conversión de un ID de mosaico a la dirección CTB en el escaneo de mosaicos del primer CTB en el mosaico es derivada de la siguiente manera:

```

for( ctbAddrTs = 0, tileIdx = 0, tileStartFlag = 1; ctbAddrTs < SubPicSizeInCtbsY; ctbAddrTs++ ) {
if( tileStartFlag ) {
    FirstCtbAddrTs[ tileIdx ] = ctbAddrTs
    tileStartFlag = 0
}
tileEndFlag = ctbAddrTs == SubPicSizeInCtbsY - 1 || TileId[ ctbAddrTs + 1 ] != TileId[ ctbAddrTs ]
if( tileEndFlag ) {
    tileIdx++
    tileStartFlag = 1
}
}

```

(6-9)

Los valores de ColumnWidthInLumaSamples[i], especificando la anchura de la i-ava columna de mosaicos en unidades de muestras de luminancia, se establecen igual a ColWidth[i] << CtbLog2SizeY para i que varía de 0 a num_tile_columns_minus1, inclusivo. Los valores de RowHeightInLumaSamples[j], especificando la altura de la j-ava fila de mosaicos en unidades de muestras de luminancia, se establecen igual a RowHeight[j] << CtbLog2SizeY para j que varía de 0 a num_tile_rows_minus1, inclusivo.

Un ejemplo de la sintaxis de conjunto de parámetros de secuencia RBSP es de la siguiente manera.

seq_parameter_set_rbsp() {	Descriptor
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if (chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
num_sub_pics_minus1	ue(v)
sub_pic_id_len_minus1	ue(v)
if (num_sub_pics_minus1 > 0)	
sub_pic_level_present_flag	u(1)
for (i = 0; i <= num_sub_pics_minus1; i++) {	
sub_pic_id[i]	u(v)

if (num_sub_pics_minus1 > 0) {	
sub_pic_treated_as_pic_flag[i]	u(1)
si (sub_pic_treated_as_pic_flag[i] && sub_pic_level_present_flag)	
sub_pic_level_idc[i]	u(8)
sub_pic_x_offset[i]	ue(v)
sub_pic_y_offset[i]	ue(v)
sub_pic_width_in_luma_samples[i]	ue(v)
sub_pic_height_in_luma_samples[i]	ue(v)
}	
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
...	

Un ejemplo de sintaxis de conjunto de parámetros de imagen RBSP es de la siguiente manera.

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_id	ue(v)
loop_filter_across_sub_pic_enabled_flag	u(1)
single_tile_in_sub_pic_flag	u(1)
if (!single_tile_in_sub_pic_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_tile_spacing_flag	u(1)
if (!uniform_tile_spacing_flag) {	
for (i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for (i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
single_tile_per_tile_group_flag	u(1)
if (!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if (rect_tile_group_flag && !single_tile_per_tile_group_flag) {	
num_tile_groups_in_sub_pic_minus1	ue(v)
for (i = 0; i <= num_tile_groups_in_sub_pic_minus1; i++) {	

if (i > 0)	
top_left_tile_idx[i]	u(v)
bottom_right_tile_idx[i]	u(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
for (i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)

Un ejemplo de la sintaxis de cabecera de grupo de mosaicos general es de la siguiente manera.

tile_group_header() {	Descriptor
tile_group_pic_parameter_set_id	ue(v)
tile_group_sub_pic_id	u(v)
if (NumTilesInSubPic > 1 !(rect_tile_group_flag && num_tile_group_in_sub_pic_minus1 == 0))	
tile_group_address	u(v)

- 5 Un ejemplo de sintaxis de unidad de árbol de codificación es de la siguiente manera.

coding_tree_unit() {	Descriptor
xCtb = ((CtbAddrInRs % SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_x_offset[Sub-PicIdx[tile_group_subpic_id]]	
yCtb = ((CtbAddrInRs / SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_y_offset[Sub-PicIdx[tile_group_subpic_id]]	
...	

Un ejemplo de semántica de conjunto de parámetros de secuencia RBSP es de la siguiente manera.

- 10 El bit_depth_chroma_minus8 especifica la profundidad de bit de las muestras de los arreglos de crominancia BitDepthC y el valor de la compensación de rango de parámetro de cuantificación de crominancia QpBdOffsetC es de la siguiente manera:

$$\text{BitDepthC} = 8 + \text{bit_depth_chroma_minus8} \quad (7-4)$$

$$\text{QpBdOffsetC} = 6 * \text{bit_depth_chroma_minus8} \quad (7-5)$$
- 15 El bit_depth_chroma_minus8 deberá estar en el rango de cero a ocho, inclusivo.

- 20 El num_sub_pics_minus1 más uno especifica el número de sub-imágenes en cada imagen codificada en el CVS. El valor de num_sub_pics_minus1 deberá estar en rango de cero a mil veinticuatro, inclusivo. El sub_pic_id_len_minus1 más uno especifica el número de bits utilizados para representar el elemento de sintaxis sub_pic_id[i] en el SPS y el elemento de sintaxis tile_group_sub_pic_id en las cabeceras de grupos de mosaicos. El valor de sub_pic_id_len_minus1 deberá estar en el rango de Techo($\text{Log2}(\text{num_sub_pic_minus1} + 1) - 1$) a nueve, inclusivo. El sub_pic_level_present_flag se establece a uno para especificar que el elemento de sintaxis sub_pic_level_idc[i] puede estar presente. El sub_pic_level_present_flag se establece a cero para especificar que el elemento de sintaxis sub_pic_level_idc[i] no está presente. El sub_pic_id[i] especifica el ID de sub-imagen de la

i-ava sub-imagen de cada imagen codificada en el CVS. La longitud de `sub_pic_id[i]` es `sub_pic_id_len_minus1 + 1` bits.

El `sub_pic_treated_as_pic_flag[i]` es establecido igual a uno para especificar que la i-ava sub-imagen de cada imagen codificada en el CVS es tratada como una imagen en el proceso de decodificación excluyendo operaciones de filtrado en-bucle. El `sub_pic_treated_as_pic_flag[i]` es establecido igual a cero para especificar que la i-ava sub-imagen de cada imagen codificada en el CVS no es tratada como una imagen en el proceso de decodificación excluyendo operaciones de filtrado en-bucle. El `sub_pic_level_idc[i]` indica el nivel al cual se adecúa la i-ava secuencia de sub-imagen, donde la i-ava secuencia de sub-imágenes consiste de únicamente las unidades NAL VCL de las sub-imágenes con ID de sub-imagen igual a `sub_pic_id[i]` en el CVS y sus unidades NAL no-VCL asociadas. El `sub_pic_x_offset[i]` especifica la compensación horizontal, en unidades de muestras de luminancia, de la muestra de luminancia de esquina superior-izquierda de la i-ava sub-imagen con relación a la muestra de luminancia de esquina superior-izquierda de cada imagen en el CVS. Cuando no está presente, el valor de `sub_pic_x_offset[i]` es inferido para que sea igual a cero. El `sub_pic_y_offset[i]` especifica la compensación vertical, en unidades de muestras de luminancia, de la muestra de luminancia de esquina superior-izquierda de la i-ava sub-imagen con relación a la muestra de luminancia de esquina superior-izquierda de cada imagen en el CVS. Cuando no está presente, el valor de `sub_pic_y_offset[i]` es inferido para que sea igual a cero. El `sub_pic_width_in_luma_samples[i]` especifica la anchura, en unidades de muestras de luminancia, de la i-ava sub-imagen de cada imagen en el CVS. Cuando la suma de `sub_pic_x_offset[i]` y `sub_pic_width_in_luma_samples[i]` es menor que `pic_width_in_luma_samples`, el valor de `sub_pic_width_in_luma_samples[i]` debiera ser un entero múltiplo de `CtbSizeY`. Cuando no está presente, el valor de `sub_pic_width_in_luma_samples[i]` es inferido para que sea igual a `pic_width_in_luma_samples`. El `sub_pic_height_in_luma_samples[i]` especifica la altura, en unidades de muestras de luminancia, de la i-ava sub-imagen para cada imagen en el CVS. Cuando la suma de `sub_pic_y_offset[i]` y `sub_pic_height_in_luma_samples[i]` es menor que `pic_height_in_luma_samples`, el valor de `sub_pic_height_in_luma_samples[i]` debiera ser un entero múltiplo de `CtbSizeY`. Cuando no está presente, el valor de `sub_pic_height_in_luma_samples[i]` es inferido para que sea igual a `pic_height_in_luma_samples`.

Para conformación de corriente de bits, aplican las siguientes restricciones. Para cualesquiera valores enteros de *i* y *j*, cuando *i* es igual a *j*, los valores de `sub_pic_id[i]` y `sub_pic_id[j]` no debieran ser los mismos. Para cualesquiera dos sub-imágenes subpicA y subpicB, cuando el ID de sub-imagen de subpicA es menor que el ID de sub-imagen de subpicB, cualquier unidad NAL de grupo de mosaicos de subpicA deberá seguir a cualquier unidad NAL de grupo de mosaicos codificada de subpicB en orden de decodificación. Las formas de las sub-imágenes deberán ser tales que cada sub-imagen, cuando sea decodificada, tenga su límite izquierdo completo y su límite superior completo consistiendo de un límite de imagen o consistiendo de límites de sub-imágenes previamente decodificadas.

La lista de `SubPicIdx[ spld ]` para valores `spld` iguales a `sub_pic_id[i]` con *i* variando de 0 a `num_sub_pics_minus1`, inclusivo, especificando la conversión de un ID de sub-imagen al índice de sub-imagen, es derivada de la siguiente manera:

```
for( i = 0; i <= num_sub_pics_minus1; i++ )
```

```
    SubPicIdx[ sub_pic_id[ i ] ] = i
```

(7-5)

El `log2_max_pic_order_cnt_lsb_minus4` especifica el valor de la variable `MaxPicOrderCntLsb` que es utilizado en el proceso de decodificación para conteo de orden de imágenes de la siguiente manera:

```
MaxPicOrderCntLsb = 2( log2_max_pic_order_cnt_lsb_minus4 + 4 )
```

(7-5)

El valor de `log2_max_pic_order_cnt_lsb_minus4` deberá estar en rango de cero a doce, inclusivo.

Un ejemplo de semántica de conjunto de parámetros de imagen RBSP es de la siguiente manera.

Cuando está presente, el valor de cada uno de los elementos de sintaxis `PPS pps_seq_parameter_id` y `loop_filter_across_tiles_enabled_flag` deberá ser el mismo en todos los PPSs referidos por las cabeceras de grupo de mosaicos de una imagen codificada. El `pps_pic_parameter_set_id` identifica el PPS para referencia por otros elementos de sintaxis. El valor de `pps_pic_parameter_set_id` deberá estar en rango de cero a sesenta y tres, inclusivo. El `pps_seq_parameter_id` especifica el valor de `sps_seq_parameter_set_id` para el SPS activo. El valor de `pps_seq_parameter_id` deberá estar en rango de cero a quince, inclusivo. El `loop_filter_across_sub_pic_enabled_flag` es establecido igual a uno para especificar que operaciones de filtración en-bucle pueden ser ejecutadas a través de los límites de la sub-imagen refiriéndose al PPS. El `loop_filter_across_sub_pic_enabled_flag` es establecido igual a cero para especificar que operaciones de filtración en-bucle no son ejecutadas a través de los límites de la sub-imagen refiriéndose al PPS.

El `single_tile_in_sub_pic_flag` es establecido igual a uno para especificar que hay solamente un mosaico en cada sub-imagen refiriéndose al PPS. El `single_tile_in_sub_pic_flag` es establecido igual a cero para especificar que hay más de un mosaico en cada sub-imagen refiriéndose al PPS. El `num_tile_columns_minus1` plus 1 especifica el

número de columnas de mosaicos que dividen la sub-imagen. El `num_tile_columns_minus1` deberá estar en rango de cero a mil veinticuatro, inclusivo. Cuando no está presente, el valor de `num_tile_columns_minus1` es inferido para que sea igual a cero. El `num_tile_rows_minus1` plus 1 especifica el número de filas de mosaicos que dividen la sub-imagen. El `num_tile_rows_minus1` deberá estar en rango de cero a mil veinticuatro. Cuando no está presente, el valor de `num_tile_rows_minus1` es inferido para que sea igual a cero. La variable `NumTilesInSubPic` es establecida igual a $(\text{num_tiles_columns_minus1} + 1) * (\text{num_tile_rows_minus1} + 1)$. Cuando `single_tile_in_sub_pic_flag` es igual a cero, `NumTilesInSubPic` deberá ser mayor que uno.

El `uniform_tile_spacing_flag` es establecido igual a uno para especificar que límites de columna de mosaicos y de igual manera límites de fila de mosaicos están distribuidos de manera uniforme a través de la sub-imagen. El `uniform_tile_spacing_flag` es establecido igual a cero para especificar que los límites de columna de mosaicos y de igual manera los límites de fila de mosaicos no están distribuidos de manera uniforme a través de la sub-imagen sino que son señalizados de manera explícita utilizando los elementos de sintaxis `tile_column_width_minus1[i]` y `tile_row_height_minus1[i]`. Cuando no está presente, el valor de `uniform_tile_spacing_flag` es inferido para que sea igual a uno. El `tile_column_width_minus1[i]` más 1 especifica la anchura de la i-ava columna de mosaicos en unidades de CTBs. El `tile_row_height_minus1[i]` más 1 especifica la altura de la i-ava fila de mosaicos en unidades de CTBs. El `single_tile_per_tile_group` es establecido igual a uno para especificar que cada grupo de mosaicos que se refiere a este PPS incluye un mosaico. El `single_tile_per_tile_group` es establecido igual a cero para especificar que un grupo de mosaicos que se refiere a este PPS puede incluir más de un mosaico.

El `rect_tile_group_flag` es establecido igual a cero para especificar que los mosaicos dentro de cada grupo de mosaicos de la sub-imagen están en orden de escaneo de trama y la información de grupo de mosaicos no es señalizada en el PPS. El `rect_tile_group_flag` es establecido igual a uno para especificar que mosaicos dentro de cada grupo de mosaicos cubren una región rectangular o cuadrada de la sub-imagen y la información de grupo de mosaicos es señalizada en el PPS. Cuando `single_tile_per_tile_group_flag` es establecido a uno `rect_tile_group_flag` es inferido para que sea igual a uno. El `num_tile_groups_in_sub_pic_minus1` plus 1 especifica el número de grupos de mosaicos en cada sub-imagen refiriéndose al PPS. El valor de `num_tile_groups_in_sub_pic_minus1` deberá estar en rango de cero a `NumTilesInSubPic` - 1, inclusivo. Cuando no está presente y `single_tile_per_tile_group_flag` es igual a uno, el valor de `num_tile_groups_in_sub_pic_minus1` es inferido para que sea igual a `NumTilesInSubPic` - 1.

El `top_left_tile_idx[i]` especifica el índice de mosaico del mosaico ubicado en la esquina superior-izquierda del i-avo grupo de mosaicos de la sub-imagen. El valor de `top_left_tile_idx[i]` no deberá ser igual al valor de `top_left_tile_idx[j]` para cualquiera i no igual a j. Cuando no está presente, el valor de `top_left_tile_idx[i]` es inferido para que sea igual a i. La longitud del elemento de sintaxis `top_left_tile_idx[i]` es $\text{Techo}(\text{Log2}(\text{NumTilesInSubPic}))$ bits. El `bottom_right_tile_idx[i]` especifica el índice de mosaico del mosaico ubicado en la esquina inferior-derecha del i-avo grupo de mosaicos de la sub-imagen. Cuando `single_tile_per_tile_group_flag` es establecido a uno, el `bottom_right_tile_idx[i]` es inferido para que sea igual a `top_left_tile_idx[i]`. La longitud del elemento de sintaxis `bottom_right_tile_idx[i]` es $\text{Techo}(\text{Log2}(\text{NumTilesInSubPic}))$ bits.

Un requerimiento de la conformación de corriente de bits es que cualquier mosaico particular solamente debiera ser incluido en un grupo de mosaicos. La variable `NumTilesInTileGroup[i]`, la cual especifica el número de mosaicos en el i-avo grupo de mosaicos de la sub-imagen, y variables relacionadas, son derivadas de la siguiente manera:

$$\text{NumTileRowsInTileGroupMinus1}[i] = \text{deltaTileIdx} / (\text{num_tile_columns_minus1} + 1)$$

$$\text{NumTileColumnsInTileGroupMinus1}[i] = \text{deltaTileIdx} \% (\text{num_tile_columns_minus1} + 1)$$

(7-33)

$$\text{NumTilesInTileGroup}[i] = (\text{NumTileRowsInTileGroupMinus1}[i] + 1) *$$

$$(\text{NumTileColumnsInTileGroupMinus1}[i] + 1)$$

El `loop_filter_across_tiles_enabled_flag` es establecido igual a uno para especificar que operaciones de filtración en-bucle pueden ser ejecutadas a través de límites de mosaico en sub-imágenes refiriéndose al PPS. El `loop_filter_across_tiles_enabled_flag` es establecido igual a cero para especificar que operaciones de filtración en-bucle no son ejecutadas a través de límites de mosaico en sub-imágenes refiriéndose al PPS. Las operaciones de filtración en-bucle incluyen operaciones del filtro de desbloqueo, el filtro de compensación adaptativa de muestra, y el filtro de bucle adaptativo. Cuando no está presente, el valor de `loop_filter_across_tiles_enabled_flag` es inferido para que sea igual a uno. El `num_ref_idx_default_active_minus1[i]` más 1, cuando i es igual a cero, especifica el valor inferido de la variable `NumRefIdxActive[0]` para P o B grupos de mosaicos con `num_ref_idx_active_override_flag` igual a 0, y, cuando i es igual a uno, especifica el valor inferido de `NumRefIdxActive[1]` para B grupos de mosaicos con `num_ref_idx_active_override_flag` igual a cero. El valor de `num_ref_idx_default_active_minus1[i]` deberá estar en rango de cero a catorce, inclusivo.

Un ejemplo de semántica de cabecera de grupo de mosaicos general es de la siguiente manera. Cuando está presente, el valor de cada uno de los elementos de sintaxis de cabecera de grupo de mosaicos `tile_group_pic_order_cnt_lsb` y `tile_group_temporal_mvp_enabled_flag` deberá ser el mismo en todas las cabeceras de grupo de mosaicos de una imagen codificada. Cuando está presente, el valor del `tile_group_pic_parameter_set_id` deberá ser el mismo en todas las cabeceras de grupo de mosaicos de una sub-imagen codificada. El `tile_group_pic_parameter_set_id` especifica el valor de `pps_pic_parameter_set_id` para el PPS en uso. El valor de `tile_group_pic_parameter_set_id` deberá estar en rango de cero a sesenta y tres, inclusivo. Es requerimiento de la conformación de la corriente de bits que el valor de `TemporalId` de la imagen actual debiera ser mayor que o igual al valor de `TemporalId` de cada PPS referido por un grupo de mosaicos de la imagen actual. El `tile_group_sub_pic_id` identifica la sub-imagen a la cual pertenece el grupo de mosaicos. La longitud de `tile_group_sub_pic_id` es `sub_pic_id_len_minus1 + 1` bits. El valor de `tile_group_sub_pic_id` deberá ser el mismo para todas las cabeceras de grupo de mosaicos de una sub-imagen codificada.

Las variables `SubPicWidthInCtbsY`, `SubPicHeightInCtbsY`, y `SubPicSizeInCtbsY` son derivadas de la siguiente manera:

$i = \text{SubPicIdx}[\text{tile_group_subpic_id}]$

$\text{SubPicWidthInCtbsY} =$

$$\text{Ceil}(\text{sub_pic_width_in_luma_samples}[i] \div \text{CtbSizeY}) \quad (7-34)$$

$\text{SubPicHeightInCtbsY} = \text{Ceil}(\text{sub_pic_height_in_luma_samples}[i] \div \text{CtbSizeY})$

$\text{SubPicSizeInCtbsY} = \text{SubPicWidthInCtbsY} * \text{SubPicHeightInCtbsY}$

Las siguientes variables son derivadas recurriendo al Proceso de conversión de escaneo de mosaico y trama CTB: la lista `ColWidth[i]` para i variando de cero a `num_tile_columns_minus1`, inclusivo, especificando la anchura de la i -ava columna de mosaicos en unidades de CTBs; la lista `RowHeight[j]` para j variando de cero a `num_tile_rows_minus1`, inclusivo, especificando la altura de la j -ava fila de mosaicos en unidades de CTBs; la lista `ColBd[i]` para i variando de cero a `num_tiles_columns_minus1 + 1`, inclusivo, especificando la ubicación del i -avo límite de columna de mosaicos en unidades de CTBs; la lista `RowBd[j]` para j variando de cero a `num_tile_rows_minus1 + 1`, inclusivo, especificando la ubicación del j -avo límite de fila de mosaicos en unidades de CTBs; la lista `CtbAddrRsATs[ctbAddrRs]` para `ctbAddrRs` variando de cero a `SubPicSizeInCtbsY - 1`, inclusivo, especificando la conversión de una dirección CTB en el escaneo de trama CTB de una sub-imagen a una dirección CTB en el escaneo de mosaico de la sub-imagen; la lista `CtbAddrTsToRs[ctbAddrTs]` para `ctbAddrTs` variando de cero a `SubPicSizeInCtbsY - 1`, inclusivo, especificando la conversión de una dirección CTB en el escaneo de mosaico de una sub-imagen a una dirección CTB en el escaneo de trama CTB de la sub-imagen; la lista `TileId[ctbAddrTs]` para `ctbAddrTs` variando de cero a `SubPicSizeInCtbsY - 1`, inclusivo, especificando la conversión de una dirección CTB en el escaneo de mosaico de una sub-imagen a un ID de mosaico; la lista `NumCtusInTile[TileIdx]` para `TileIdx` variando de cero a `NumTilesInSubPic - 1`, inclusivo, especificando la conversión de un índice de mosaico al número de CTUs en el mosaico; la lista `FirstCtbAddrTs[TileIdx]` para `TileIdx` variando de cero a `NumTilesInSubPic - 1`, inclusivo, especificando la conversión de un ID de mosaico a la dirección CTB en el escaneo de mosaico del primer CTB en el mosaico; la lista `ColumnWidthInLumaSamples[i]` para i variando de cero a `num_tile_columns_minus1`, inclusivo, especificando la anchura de la i -ava columna de mosaicos en unidades de muestras de luminancia; y la lista `RowHeightInLumaSamples[j]` para j variando de cero a `num_tile_rows_minus1`, inclusivo, especificando la altura de la j -ava fila de mosaicos en unidades de muestras de luminancia.

Los valores de `ColumnWidthInLumaSamples[i]` para i variando de cero a `num_tile_columns_minus1`, inclusivo, y `RowHeightInLumaSamples[j]` para j variando de cero a `num_tile_rows_minus1`, inclusivo, deberán ser mayores que cero. Las variables `SubPicLeftBoundaryPos`, `SubPicTopBoundaryPos`, `SubPicRightBoundaryPos` y `SubPicBotBoundaryPos` son derivadas de la siguiente manera:

$i = \text{SubPicIdx}[\text{tile_group_subpic_id}]$

$\text{if}(\text{sub_pic_treated_as_pic_flag}[i]) \{$

$\text{SubPicLeftBoundaryPos} = \text{sub_pic_x_offset}[i]$

$\text{SubPicRightBoundaryPos} = \text{SubPicLeftBoundaryPos} + \text{sub_pic_width_in_luma_samples}[i] - 1$

$\text{SubPicTopBoundaryPos} = \text{sub_pic_y_offset}[i]$

$\text{SubPicBotBoundaryPos} = \text{SubPicTopBoundaryPos} + \text{sub_pic_height_in_luma_samples}[i] - 1 \quad (7-34)$

$\}$

Para cada mosaico, con índice $i = 0$. `NumTilesInSubPic - 1` en la sub-imagen actual, las variables `TileLeftBoundaryPos[i]`, `TileTopBoundaryPos[i]`, `TileRightBoundaryPos[i]`, y `TileBotBoundaryPos[i]` son derivadas de la siguiente manera:

```

tileColIdx = i % ( num_tile_columns_minus1 + 1 )
tileRowIdx = i / ( num_tile_columns_minus1 + 1 )
TileLeftBoundaryPos[ i ] = SubPicLeftBoundaryPos + ( ColBd[ tileColIdx ] << CtbLog2SizeY )
TileRightBoundaryPos[ i ] = SubPicLeftBoundaryPos +
    ( ( ColBd[ tileColIdx ] + ColWidth[ tileColIdx ] ) << CtbLog2SizeY ) - 1
if( TileRightBoundaryPos[ i ] > pic_width_in_luma_samples - 1 )
    TileRightBoundaryPos[ i ] = pic_width_in_luma_samples - 1
TileTopBoundaryPos[ i ] = SubPicTopBoundaryPos + ( RowBd[ tileRowIdx ] << CtbLog2SizeY )
TileBotBoundaryPos[ i ] = SubPicTopBoundaryPos +
    ( ( RowBd[ tileRowIdx ] + RowHeight[ tileRowIdx ] ) << CtbLog2SizeY ) - 1
if( TileBotBoundaryPos[ i ] > pic_height_in_luma_samples - 1 )
    TileBotBoundaryPos[ i ] = pic_height_in_luma_samples - 1

```

- 5 El tile_group_address especifica la dirección de mosaico del primer mosaico en el grupo de mosaicos. Cuando no está presente, el valor de tile_group_address es inferido para que sea igual a cero. Si rect_tile_group_flag es igual a 0, aplica lo siguiente: la dirección de mosaico es el ID de mosaico; la longitud de tile_group_address es $\text{Techo}(\text{Log2}(\text{NumTilesInSubPic}))$ bits; y el valor de tile_group_address deberá estar en rango de cero a $\text{NumTilesInSubPic} - 1$, inclusivo. De otra manera (rect_tile_group_flag es igual a uno), aplica lo siguiente: la
- 10 dirección de mosaico es el índice de grupo de mosaicos del grupo de mosaicos entre grupos de mosaicos en la sub-imagen; la longitud de tile_group_address es $\text{Techo}(\text{Log2}(\text{num_tile_groups_in_sub_pic_minus1} + 1))$ bits; y el valor de tile_group_address deberá estar en el rango de 0 a num_tile_groups_in_sub_pic_minus1, inclusivo.

- 15 Un requerimiento de la conformación de corriente de bits es que apliquen las siguientes restricciones. El valor de tile_group_address no deberá ser igual al valor de tile_group_address de ninguna otra unidad NAL de grupo de mosaicos codificado de la misma sub-imagen codificada. El grupos de mosaicos de una sub-imagen deberá estar en orden de incremento de sus valores tile_group_address. Las formas de los grupos de mosaicos de una sub-imagen debieran ser tales que cada mosaico, cuando es decodificado, debiera tener su límite izquierdo completo y su límite superior completo consistiendo de un límite de sub-imagen o consistiendo de límites de mosaicos previamente decodificados.
- 20

- 25 El num_tiles_in_tile_group_minus1, cuando está presente, especifica el número de mosaicos en el grupo de mosaicos menos uno. El valor de num_tiles_in_tile_group_minus1 deberá estar en rango de cero a $\text{NumTilesInSubPic} - 1$, inclusivo. Cuando no está presente, el valor de num_tiles_in_tile_group_minus1 es inferido para que sea igual a cero. La variable NumTilesInCurrTileGroup, la cual especifica el número de mosaicos en el grupo de mosaicos actual, y TgTileIdx[i], la cual especifica el índice de mosaico del i-avo mosaico en el grupo de mosaicos actual, son derivadas de la siguiente manera:

```

if( rect_tile_group_flag ) {
    NumTilesInCurrTileGroup = NumTilesInTileGroup[ tile_group_address ]
    tileIdx = top_left_tile_idx[ tile_group_address ]
    for( j = 0, tIdx = 0; j <
        NumTileRowsInTileGroupMinus1[ tile_group_address ] + 1;
        j++, tileIdx += num_tile_columns_minus1 + 1 )
        for( i = 0, currTileIdx = tileIdx; i < NumTileColumnsInTileGroupMinus1[ tile_group_address ] + 1;
            i++, currTileIdx++, tIdx++ )
            TgTileIdx[ tIdx ] = currTileIdx
    } else {
        NumTilesInCurrTileGroup = num_tiles_in_tile_group_minus1 + 1
        TgTileIdx[ 0 ] = tile_group_address
        for( i = 1; i < NumTilesInCurrTileGroup; i++ )
            TgTileIdx[ i ] = TgTileIdx[ i - 1 ] + 1
    }

```

- 30 El tile_group_type especifica el tipo de codificación del grupo de mosaicos.

Un proceso de derivación ejemplar para predicción de vector de movimiento de luminancia temporal es de la siguiente manera. Las variables mvLXCol y availableFlagLXCol son derivadas de la siguiente manera: Si tile_group_temporal_mvp_enabled_flag es igual a cero, ambos componentes de mvLXCol son establecidos igual a cero y availableFlagLXCol es establecido igual a cero. De otra manera (tile_group_temporal_mvp_enabled_flag es igual a uno), aplican los siguientes pasos ordenados: El vector de movimiento colocado inferior derecha, y las ubicaciones de muestra de límite inferior y derecha son derivadas de la siguiente manera:

xColBr = xCb + cbWidth (8-414)

yColBr = yCb + cbHeight (8-415)

rightBoundaryPos = sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]] ?

SubPicRightBoundaryPos : pic_width_in_luma_samples - 1 (8-415)

botBoundaryPos = sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]] ?

SubPicBotBoundaryPos : pic_height_in_luma_samples - 1 (8-415)

Si yCb >> CtbLog2SizeY es igual a yColBr >> CtbLog2SizeY, yColBr es menor que o igual a botBoundaryPos y xColBr es menor que o igual a rightBoundaryPos, aplica lo siguiente: La variable colCb especifica el bloque de codificación de luminancia cubriendo la ubicación modificada proporcionada por ((xColBr >> 3) << 3, (yColBr >> 3) << 3) dentro de la imagen colocada especificada por ColPic. La ubicación de luminancia (xColCb, yColCb) es establecida igual a la muestra superior-izquierda del bloque de codificación de luminancia colocado especificado por colCb con relación a la muestra de luminancia superior-izquierda de la imagen colocada especificada por ColPic. El proceso de derivación para vectores de movimiento colocados es consultado con currCb, colCb, (xColCb, yColCb), refIdxLX y sbFlag establecidos igual a cero como entradas, y la salida es asignada a mvLXCol y availableFlagLXCol. De otra manera, ambos componentes de mvLXCol son establecidos igual a cero y availableFlagLXCol es establecido igual a cero.

Un ejemplo de proceso de interpolación bilineal de muestra de luminancia es de la siguiente manera. Las ubicaciones de luminancia en unidades de muestra completa (xInti, yInti) son derivadas de la siguiente manera para i = 0..1. Si sub_pic_treated_as_pic_flag[Sub-PicIdx[tile_group_subpic_id]] es igual a 1, aplica lo siguiente:

xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i) (8-458)

yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i) (8-458)

De otra manera (sub_pic_treated_as_pic_flag[Sub-PicIdx[tile_group_subpic_id]] es igual a 0), aplica lo siguiente:
xInti = sps_ref_wraparound_enabled_flag ?

ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, (xIntL + i)) :

(8-459)

Clip3(0, picW - 1, xIntL + i)

yInti = Clip3(0, picH - 1, yIntL + i) (8-460)

Un ejemplo de proceso de filtración de interpolación de 8 derivaciones de muestra de luminancia es de la siguiente manera. Las ubicaciones de luminancia en unidades de muestra completa (xInti, yInti) son derivadas de la siguiente manera para i = 0..7. Si sub_pic_treated_as_pic_flag[Sub-PicIdx[tile_group_subpic_id]] es igual a uno, aplica lo siguiente:

xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i - 3) (8-830)

yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i - 3) (8-830)

De otra manera (sub_pic_treated_as_pic_flag[Sub-PicIdx[tile_group_subpic_id]] es igual a 0), aplica lo siguiente:
xInti = sps_ref_wraparound_enabled_flag ?

ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, xIntL + i - 3) :

(8-831)

Clip3(0, picW - 1, xIntL + i - 3)

yInti = Clip3(0, picH - 1, yIntL + i - 3) (8-832)

Un proceso de interpolación de muestra de crominancia ejemplar es de la siguiente manera. La variable xOffset es establecida igual a $(\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY} / \text{SubWidthC}$. Las ubicaciones de crominancia en unidades de muestra completa (xInti, yInti) son derivadas de la siguiente manera para $i = 0..3$:

- 5 Si `sub_pic_treated_as_pic_flag[ SubPicIdx[ tile_group_subpic_id ] ]` es igual a uno, aplica lo siguiente:
`xInti = Clip3( SubPicLeftBoundaryPos / SubWidthC, SubPicRightBoundaryPos / SubWidthC, xIntL + i )`

(8-844)

`yInti = Clip3( SubPicTopBoundaryPos / SubHeightC, SubPicBotBoundaryPos / SubHeightC, yIntL + i )`

(8-844)

- 10 De otra manera (`sub_pic_treated_as_pic_flag[ SubPicIdx[ tile_group_subpic_id ] ]` es igual a 0), aplica lo siguiente:
`xInti = sps_ref_wraparound_enabled_flag ? ClipH( xOffset, picWC, xIntC + i - 1 ) :` (8-845)

`Clip3( 0, picWC - 1, xIntC + i - 1 )`

`yInti = Clip3( 0, picHC - 1, yIntC + i - 1 )`

(8-846)

- 15 Un ejemplo de proceso de filtro de desbloqueo es de la siguiente manera. El proceso de filtro de desbloqueo es aplicado a todos los bordes de sub-bloque de codificación y bordes de bloque de transformada de una imagen, excepto los siguientes tipos de bordes: bordes que están en el límite de la imagen; bordes que coinciden con límites de sub-imágenes para los cuales `loop_filter_across_sub_pic_enabled_flag` es igual a cero; bordes que coinciden con límites de mosaicos para los cuales `loop_filter_across_tiles_enabled_flag` es igual a cero; bordes que coinciden con los límites superior o izquierdo de o dentro de grupos de mosaicos con `tile_group_deblocking_filter_disabled_flag` igual a uno; bordes que no corresponden a límites de rejilla de muestra 8x8 del componente considerado; bordes dentro de componentes de crominancia para los cuales ambos lados del borde utilizan inter predicción; bordes de bloques de transformada de crominancia que no son los bordes de la unidad de transformada asociada; y bordes a través de los bloques de transformada de luminancia de una unidad de codificación que tiene un valor IntraSubParticionesDivisión no igual a ISP_NO_DIVISIÓN.

Un ejemplo de proceso de filtro de desbloqueo para una dirección es de la siguiente manera. Para cada unidad de codificación con anchura de bloque de codificación $\log_2\text{CbW}$, altura de bloque de codificación $\log_2\text{CbH}$ y ubicación de muestra superior-izquierda del bloque de codificación (xCb, yCb), cuando `EdgeType` es igual a `EDGE_VER` y $\text{xCb} \% 8$ es igual a cero o cuando `EdgeType` es igual a `EDGE_HOR` y $\text{yCb} \% 8$ es igual a cero, los bordes son filtrados por los siguientes pasos ordenados. La anchura de bloque de codificación nCbW es establecida igual a $1 \ll \log_2\text{CbW}$ y la altura de bloque de codificación nCbH es establecida igual a $1 \ll \log_2\text{CbH}$. La variable `filterEdgeFlag` es derivada de la siguiente manera. Si `EdgeType` es igual a `EDGE_VER` y una o más de las siguientes condiciones son verdaderas, `filterEdgeFlag` es establecido igual a cero. El límite izquierdo del bloque de codificación actual es el límite izquierdo de la imagen. El límite izquierdo del bloque de codificación actual es el límite izquierdo o derecho de la sub-imagen y `loop_filter_across_sub_pic_enabled_flag` es igual a cero. El límite izquierdo del bloque de codificación actual es el límite izquierdo del mosaico y `loop_filter_across_tiles_enabled_flag` es igual a cero. De otra manera si `EdgeType` es igual a `EDGE_HOR` y una o más de las siguientes condiciones son verdaderas, la variable `filterEdgeFlag` es establecida igual a 0. El límite superior del bloque de codificación de luminancia actual es el límite superior de la imagen. El límite superior del bloque de codificación actual es el límite superior o inferior de la sub-imagen y `loop_filter_across_sub_pic_enabled_flag` es igual a cero. El límite superior del bloque de codificación actual es el límite superior del mosaico y `loop_filter_across_tiles_enabled_flag` es igual a cero. De otra manera, `filterEdgeFlag` es establecido igual a uno.

- 45 Un ejemplo de proceso de modificación CTB es de la siguiente manera. Para todas las ubicaciones de muestra (xSi, ySj) y (xYi, yYj) con $i = 0..n\text{CtbSw} - 1$ y $j = 0..n\text{CtbSh} - 1$, dependiendo de los valores de `pcm_loop_filter_disabled_flag`, `pcm_flag[ xYi ][ yYj ]` y `cu_transquant_bypass_flag` de la unidad de codificación la cual incluye el bloque de codificación que cubre `recPicture[ xSi ][ ySj ]`, aplica lo siguiente. Si una o más de las siguientes condiciones para todas las ubicaciones de muestra (xSik, ySjk') y (xYik', yYjk') con $k = 0..1$ son verdaderas, `bordeldx` es establecido igual a 0. La muestra en la ubicación (xSik, ySjk') está fuera de los límites de imagen. La muestra en la ubicación (xSik, ySjk') pertenece a una sub-imagen diferente y `loop_filter_across_sub_pic_enabled_flag` en el grupo de mosaicos al cual pertenece la muestra `recPicture[ xSi ][ ySj ]` es igual a 0. El `loop_filter_across_tiles_enabled_flag` es igual a cero y la muestra en la ubicación (xSik, ySjk') pertenece a un mosaico diferente.

- 55 Un ejemplo de proceso de filtración de bloque de árbol de codificación para muestras de luminancia es de la siguiente manera. Para la derivación de las muestras de luminancia reconstruidas filtradas `alfPictureL[ x ][ y ]`, cada muestra de luminancia reconstruida dentro del bloque de árbol de codificación de luminancia actual

recPicture[x][y] es filtrada de la siguiente manera con $x, y = 0..CtbSizeY - 1$. Las ubicaciones (hx, vy) para cada una de las muestras de luminancia correspondientes (x, y) dentro del arreglo determinado recPicture de muestras de luminancia son derivadas de la siguiente manera. Si loop_filter_across_tiles_enabled_flag para el mosaico mosaicoA que contiene la muestra de luminancia en la ubicación (hx, vy) es igual a cero, asumir que la variable

5 TileIdx es el índice de mosaico de mosaico A, aplica lo siguiente:

$$hx = Clip3(TileLeftBoundaryPos[tileIdx], TileRightBoundaryPos[tileIdx], xCtb + x) \quad (8-1140)$$

$$vy = Clip3(TileTopBoundaryPos[tileIdx], TileBotBoundaryPos[tileIdx], yCtb + y) \quad (8-1141)$$

De otra manera, si loop_filter_across_sub_pic_enabled_flag es la sub-imagen que contiene la muestra de luminancia en la ubicación (hx, vy) es igual a cero, aplica lo siguiente:

10
$$hx = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xCtb + x) \quad (8-1140)$$

$$vy = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yCtb + y) \quad (8-1141)$$

15 De otra manera, aplica lo siguiente:

$$hx = Clip3(0, pic_width_in_luma_samples - 1, xCtb + x) \quad (8-1140)$$

$$vy = Clip3(0, pic_height_in_luma_samples - 1, yCtb + y) \quad (8-1141)$$

20 Un proceso de derivación ejemplar para transpuesta ALF e índice de filtro para muestras de luminancia es de la siguiente manera. Las ubicaciones (hx, vy) para cada una de las muestras de luminancia correspondientes (x, y) dentro del arreglo determinado recPicture de muestras de luminancia son derivadas de la siguiente manera. Si loop_filter_across_tiles_enabled_flag para el mosaico A que contiene la muestra de luminancia en la ubicación (hx, vy) es igual a cero, asumir que el TileIdx es el índice de mosaico de mosaicoA, aplica lo siguiente:

25
$$hx = Clip3(TileLeftBoundaryPos[tileIdx], TileRightBoundaryPos[tileIdx], x) \quad (8-1140)$$

$$vy = Clip3(TileTopBoundaryPos[tileIdx], TileBotBoundaryPos[tileIdx], y) \quad (8-1141)$$

De otra manera, si loop_filter_across_sub_pic_enabled_flag para la sub-imagen que contiene la muestra de luminancia en la ubicación (hx, vy) es igual a cero, aplica lo siguiente:

30
$$hx = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, x) \quad (8-1140)$$

$$vy = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, y) \quad (8-1141)$$

35 De otra manera, aplica lo siguiente:

$$hx = Clip3(0, pic_width_in_luma_samples - 1, x) \quad (8-1145)$$

$$vy = Clip3(0, pic_height_in_luma_samples - 1, y) \quad (8-1146)$$

40 Un ejemplo de proceso de filtración de bloque de árbol de codificación para muestras de crominancia es de la siguiente manera. Para la derivación de las muestras de crominancia reconstruidas filtradas alfPicture[x][y], cada muestra de crominancia reconstruida dentro del bloque de árbol de codificación de crominancia actual recPicture[x][y] es filtrada de la siguiente manera con $x, y = 0..CtbSizeC - 1$. Las ubicaciones (hx, vy) para cada una de las muestras de crominancia correspondientes (x, y) dentro del arreglo determinado recPicture de muestras de crominancia son derivadas de la siguiente manera. Si loop_filter_across_tiles_enabled_flag para el mosaico mosaicoA que contiene la muestra de crominancia en la ubicación (hx, vy) es igual a cero, asumir que TileIdx es el índice de mosaico of mosaico A, aplica lo siguiente:

45
$$hx = Clip3(TileLeftBoundaryPos[tileIdx] / SubWidthC, \quad (8-1140)$$

$$TileRightBoundaryPos[tileIdx] / SubWidthC, xCtb + x)$$

50
$$vy = Clip3(TileTopBoundaryPos[tileIdx] / SubWidthC, \quad (8-1141)$$

$$TileBotBoundaryPos[tileIdx] / SubWidthC, yCtb + y)$$

De otra manera, si loop_filter_across_sub_pic_enabled_flag para la sub-imagen que contiene la muestra de crominancia en la ubicación (hx, vy) es igual a cero, aplica lo siguiente:

$hx = \text{Clip3}(\text{SubPicLeftBoundaryPos} / \text{SubWidthC},$
 $\text{SubPicRightBoundaryPos} / \text{SubWidthC}, xCtb + x)$ (8-1140)

$vy = \text{Clip3}(\text{SubPicTopBoundaryPos} / \text{SubWidthC},$
 $\text{SubPicBotBoundaryPos} / \text{SubWidthC}, yCtb + y)$ (8-1141)

De otra manera, aplica lo siguiente:

$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, xCtb + x)$ (8-1177)

$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, yCtb + y)$ (8-1178)

La variable de suma es derivada de la siguiente manera:

$\text{sum} = \text{AlfCoeffC}[0] * (\text{recPicture}[hx, vy + 2] + \text{recPicture}[hx, vy - 2]) +$
 $\text{AlfCoeffC}[1] * (\text{recPicture}[hx + 1, vy + 1] + \text{recPicture}[hx - 1, vy - 1]) +$
 $\text{AlfCoeffC}[2] * (\text{recPicture}[hx, vy + 1] + \text{recPicture}[hx, vy - 1]) +$ (8-1179)
 $\text{AlfCoeffC}[3] * (\text{recPicture}[hx - 1, vy + 1] + \text{recPicture}[hx + 1, vy - 1]) +$
 $\text{AlfCoeffC}[4] * (\text{recPicture}[hx + 2, vy] + \text{recPicture}[hx - 2, vy]) +$
 $\text{AlfCoeffC}[5] * (\text{recPicture}[hx + 1, vy] + \text{recPicture}[hx - 1, vy]) +$
 $\text{AlfCoeffC}[6] * \text{recPicture}[hx, vy]$

$\text{sum} = (\text{sum} + 64) \gg 7$ (8-1180)

La muestra de imagen de crominancia reconstruida filtrada modificada $\text{alfPicture}[xCtbC + x][yCtbC + y]$ es derivada de la siguiente manera:

$\text{alfPicture}[xCtbC + x][yCtbC + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum})$ (8-1181)

La figura 12 es un diagrama esquemático de un dispositivo de codificación de vídeo 1200 ejemplar. El dispositivo de codificación de vídeo 1200 es conveniente para implementar los ejemplos/realizaciones divulgadas tal como aquí se describe. El dispositivo de codificación de vídeo 1200 comprende puertos corriente abajo 1220, puertos corriente arriba 1250, y/o unidades de transceptor (Tx/Rx) 1210, incluyendo transmisores y/o receptores para comunicar datos corriente arriba y/o corriente abajo sobre una red. El dispositivo de codificación de vídeo 1200 también incluye un procesador 1230 incluyendo una unidad lógica y/o unidad de procesamiento central (CPU) para procesar los datos y una memoria 1232 para almacenar los datos. El dispositivo de codificación de vídeo 1200 también pueden comprender componentes eléctricos, componentes óptico-a-eléctrico (OE), componentes eléctrico-a-óptico (EO), y/o componentes de comunicación inalámbrica acoplados a los puertos corriente arriba 1250 y/o puertos corriente abajo 1220 para comunicación de datos a través de redes de comunicación eléctricas, ópticas o inalámbricas. El dispositivo de codificación de vídeo 1200 también puede incluir dispositivos de entrada y/o salida (I/O) 1260 para comunicar datos hacia y desde un usuario. Los dispositivos I/O 1260 pueden incluir dispositivos de salida tales como una pantalla para desplegar datos de vídeo, altavoces para emitir datos de audio, etc. Los dispositivos I/O 1260 también pueden incluir dispositivos de entrada, tales como un teclado, ratón, rueda de desplazamiento, etc., y/o interfaces correspondientes para interactuar con dichos dispositivos de salida.

El procesador 1230 es implementado por hardware y software. El procesador 1230 puede ser implementado como uno o más chips CPU, núcleos (por ejemplo, como un procesador multi-núcleo), arreglos de compuerta programable en campo (FPGAs), circuitos integrados de aplicación específica (ASICs), y procesadores de señal digital (DSPs). El procesador 1230 está en comunicación con los puertos corriente abajo 1220, Tx/Rx 1210, los puertos corriente arriba 1250, y la memoria 1232. El procesador 1230 comprende un módulo de codificación 1214. El módulo de codificación 1214 implementa las realizaciones divulgadas aquí descritas, tal como los métodos 100, 1300, y 1400, los cuales pueden emplear un filtro en-bucle 1000, una corriente de bits 1100, una imagen 500, y/o bloques actuales 801 y/o 901 los cuales pueden ser codificados de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700 con base en una lista de candidatos generada de acuerdo con un patrón 900. El módulo de codificación 1214 también puede implementar cualquier otro método/mecanismo aquí descrito. Además, el módulo de codificación 1214 puede implementar un sistema de código 200, un codificador 300, y/o un decodificador 400. Por ejemplo, el módulo de codificación 1214 puede implementar la primera, segunda, tercera, cuarta, quinta y/o sexta implementaciones ejemplares como se describió antes. Por lo tanto, módulo de codificación 1214 ocasiona que el dispositivo de codificación de vídeo 1200 proporcione funcionalidad adicional y/o eficiencia de codificación cuando se codifican datos de vídeo. Debido a esto, el módulo de codificación 1214 mejora la funcionalidad del dispositivo de

codificación de vídeo 1200 así como problemas de direcciones que son específicos para la técnica de codificación de vídeo. Además, el módulo de codificación 1214 efectúa una transformación del dispositivo de codificación de vídeo 1200 a un estado diferente. Alternativamente, el módulo de codificación 1214 puede ser implementado como instrucciones almacenadas en la memoria 1232 y ejecutadas a través del procesador 1230 (por ejemplo, como un producto de programa de computadora almacenado en un medio no transitorio).

La memoria 1232 comprende uno o más tipos de memoria tales como discos, unidades de cinta, unidades de estado sólido, memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), memoria flash, memoria de contenido direccionable ternaria (TCAM), memoria de acceso aleatorio estática (SRAM), etc. La memoria 1232 puede ser utilizada como dispositivo de almacenamiento de datos de sobreflujo, para almacenar programas cuando dichos programas son seleccionados para ejecución, y para almacenar instrucciones y datos que son leídos durante la ejecución del programa.

La figura 13 es un gráfico de flujo de un método ejemplar 1300 de codificación de una secuencia de vídeo en una corriente de bits, tal como la corriente de bits 1100, direccionando un segmento, tal como el segmento 515, con relación a una sub-imagen, tal como la sub-imagen 510. El método 1300 puede ser empleado por un codificador, tal como un sistema de códec 200, un codificador 300, y/o un dispositivo de codificación de vídeo 1200 cuando se ejecuta el método 100 para codificar un bloque actual 801 y/o 901 de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700 empleando el filtro en-bucle 1000 y/o una lista de candidatos generada de acuerdo con el patrón 900.

El método 1300 puede comenzar cuando un codificador recibe una secuencia de vídeo incluyendo una pluralidad de imágenes y determina codificar esa secuencia de vídeo en una corriente de bits, por ejemplo con base en la entrada de usuario. En el paso 1301, el codificador divide una imagen en una sub-imagen y la sub-imagen en un segmento.

En el paso 1303, el codificador determina una dirección de segmento para el segmento. Específicamente, la dirección de segmento incluye un índice de segmento de nivel de sub-imagen del segmento cuando el segmento es un segmento rectangular o cuadrado. Un índice de nivel de sub-imagen es un índice que está basado en la posición de segmento en la sub-imagen en lugar de basarse en la posición de segmento en la imagen. Además, el codificador determina un ID de sub-imagen de la sub-imagen que contiene el segmento. En algunos ejemplos, el segmento puede ser un segmento rectangular o cuadrado cuando un `rect_slice_flag` es establecido igual a uno. El `rect_slice_flag` puede ser codificado en un SPS y/o PPS.

En el paso 1305, el codificador codifica la dirección de segmento y el ID de sub-imagen en la corriente de bits. Por ejemplo, el codificador puede codificar una cabecera de segmento en la corriente de bits. La cabecera de segmento está asociada con el segmento y contiene parámetros utilizados para codificar/decodificar el segmento. Por ejemplo, el segmento puede hacer referencia a la cabecera de segmento. La dirección de segmento y el ID de sub-imagen pueden ser codificados en la cabecera de segmento en la corriente de bits. En algunos ejemplos, la dirección de segmento es codificada en una variable `slice_address`, y el ID de sub-imagen es codificado en una variable `slice_subpic_id`.

En el paso 1307, el codificador codifica el segmento de la sub-imagen en la corriente de bits. El codificador también puede codificar la sub-imagen completa y/o la imagen completa en la corriente de bits. En el paso 1309, el codificador también puede codificar un SPS en la corriente de bits. Por ejemplo, el codificador puede codificar una descripción de varias sub-imágenes en el SPS. El ID de sub-imagen puede hacer referencia a la descripción de la sub-imagen relevante en el SPS. En algunos ejemplos, la descripción de la sub-imagen incluye una anchura de la sub-imagen en unidades de CTBs, una altura de la sub-imagen en unidades de CTBs, y una compensación de la sub-imagen en unidades de CTBs especificadas como una posición vertical de una CTU superior izquierda de la sub-imagen y una posición horizontal de la CTU superior izquierda de la sub-imagen. En el paso 1311, el codificador almacena la corriente de bits para comunicación hacia un decodificador.

La figura 14 es un diagrama de flujo de un método 1400 ejemplar de decodificación de una secuencia de vídeo de una corriente de bits, tal como la corriente de bits 1100, cuando un segmento, tal como el segmento 515, es direccionado con relación a una sub-imagen, tal como la sub-imagen 510. El método 1400 puede ser empleado por un decodificador, tal como un sistema de códec 200, un decodificador 400, y/o un dispositivo de codificación de vídeo 1200 cuando se ejecuta el método 100 para decodificar un bloque actual 801 y/o 901 de acuerdo con la inter-predicción unidireccional 600 y/o la inter-predicción bidireccional 700 empleando el filtro en-bucle 1000 y/o una lista de candidatos generada de acuerdo con el patrón 900.

El método 1400 puede comenzar cuando un decodificador comienza a recibir una corriente de bits de datos codificados que representan una secuencia de vídeo, por ejemplo como un resultado del método 1300. En el paso 1401, el decodificador recibe una corriente de bits. La corriente de bits puede comprender un SPS, un PPS, una

cabecera de segmento, y una sub-imagen que contiene un segmento. En algunos ejemplos, la corriente de bits también puede incluir una imagen completa incluyendo una pluralidad de sub-imágenes.

En el paso 1403, el decodificador obtiene una dirección de segmento. La dirección de segmento incluye un índice de segmento de nivel de sub-imagen del segmento cuando el segmento es un segmento rectangular o cuadrado. Un índice de nivel de sub-imagen es un índice que está basado en la posición de segmento en la sub-imagen en lugar de basarse en la posición de segmento en la imagen. Un índice de nivel de sub-imagen es un índice que está basado en la posición de segmento en la sub-imagen en lugar de basarse en la posición de segmento en la imagen. Además, el decodificador puede obtener un ID de sub-imagen de la sub-imagen que contiene el segmento. Por ejemplo, la cabecera de segmento puede estar asociada con el segmento, y puede contener parámetros utilizados para codificar/decodificar el segmento. Por ejemplo, el segmento puede hacer referencia a la cabecera de segmento. La dirección de segmento y el ID de sub-imagen se pueden obtener de la cabecera de segmento en la corriente de bits. En algunos ejemplos, la dirección de segmento está codificada en una variable `slice_address`, y el ID de sub-imagen está codificado en una variable `slice_subpic_id`. En algunos ejemplos, el segmento puede ser un segmento rectangular o cuadrado cuando un `rect_slice_flag` es establecido igual a uno. El `rect_slice_flag` se puede obtener del SPS y/o PPS.

En el paso 1405, el decodificador puede obtener una descripción de la sub-imagen del SPS. Por ejemplo, el ID de sub-imagen puede hacer referencia a la sub-imagen asociada, y por lo tanto puede hacer referencia a la descripción de la sub-imagen en el SPS. En algunos ejemplos, la descripción de la sub-imagen incluye una anchura de la sub-imagen en unidades de CTBs, una altura de la sub-imagen en unidades de CTBs, y una compensación de la sub-imagen en CTBs especificados como una posición vertical de una CTU superior izquierda de la sub-imagen y una posición horizontal de la CTU superior izquierda de la sub-imagen.

En el paso 1407, el decodificador puede colocar el segmento en la sub-imagen en base en la dirección de segmento, ID de sub-imagen, y la descripción de la sub-imagen del SPS. En el paso 1409, el decodificador decodifica el segmento de la sub-imagen con base en la dirección de segmento. El decodificador entonces puede reenviar el segmento para despliegue como parte de una secuencia de vídeo decodificada.

La figura 15 es un diagrama esquemático de un sistema ejemplar 1500 para codificar una secuencia de vídeo de imágenes en una corriente de bits, tal como corriente de bits 1100, cuando un segmento, tal como el segmento 515, es direccionado con relación a una sub-imagen, tal como la sub-imagen 510. El sistema 1500 puede ser implementado por un codificador y un decodificador tal como un sistema de códec 200, un codificador 300, un decodificador 400, y/o un dispositivo de codificación de vídeo 1200. Además, el sistema 1500 puede ser empleado cuando se implementa el método 100, 1300, y/o 1400 para codificar un bloque actual 801 y/o 901 de acuerdo con inter-predicción unidireccional 600 y/o inter-predicción bidireccional 700 empleando filtro en-bucle 1000 y/o en una lista de candidatos generada de acuerdo con el patrón 900.

El sistema 1500 incluye un codificador de vídeo 1502. El codificador de vídeo 1502 comprende un módulo de división 1505 para dividir una sub-imagen en un segmento. El codificador de vídeo 1502 además comprende un módulo de determinación 1506 para determinar una dirección de segmento incluyendo un índice de segmento de nivel de sub-imagen del segmento cuando el segmento es un segmento rectangular o cuadrado. El codificador de vídeo 1502 además comprende un módulo de codificación 1507 para codificar la dirección de segmento en una corriente de bits. El módulo de codificación 1507 además es para codificar el segmento de la sub-imagen en la corriente de bits. El codificador de vídeo 1502 además comprende un módulo de almacenamiento 1508 para almacenar la corriente de bits para comunicación hacia un decodificador. El codificador de vídeo 1502 además comprende un módulo de transmisión 1509 para transmitir la corriente de bits hacia el decodificador de vídeo 1510. El codificador de vídeo 1502 además puede ser configurado para ejecutar cualquiera de los pasos del método 1300.

El sistema 1500 también incluye un decodificador de vídeo 1510. El decodificador de vídeo 1510 comprende un módulo de recepción 1511 para recibir una corriente de bits que comprende una sub-imagen que contiene un segmento. El decodificador de vídeo 1510 además comprende un módulo de obtención 1512 para obtener una dirección de segmento incluyendo un índice de segmento de nivel de sub-imagen del segmento cuando el segmento es un segmento rectangular o cuadrado. El decodificador de vídeo 1510 además comprende un módulo de decodificación 1513 para decodificar el segmento de la sub-imagen con base en la dirección de segmento. El decodificador de vídeo 1510 además comprende un módulo de reenvío 1515 para reenviar el segmento para despliegue como parte de una secuencia de vídeo decodificada. El decodificador de vídeo 1510 además se puede configurar para ejecutar cualquiera de los pasos del método 1400.

Un primer componente está directamente acoplado a un segundo componente cuando no hay componentes intermedios, excepto por una línea, una traza, u otro medio entre el primer componente y el segundo componente. El primer componente es indirectamente acoplado al segundo componente cuando hay componentes intermedios diferentes a una línea, una traza, u otro medio entre el primer componente y el segundo componente. El término

"acoplado" y sus variantes incluyen tanto acoplado directamente como acoplado indirectamente. El uso del término "aproximadamente" significa un rango que incluye $\pm 10\%$ del número subsiguiente a menos que se indique de otra manera.

- 5 También se debiera entender que los pasos de los métodos ejemplares aquí establecidos no necesariamente requieren ser ejecutados en el orden descrito, y el orden de los pasos de dichos métodos debiera ser entendido como simplemente ejemplar. De igual manera, pasos adicionales pueden ser incluidos en dichos métodos, y algunos pasos se pueden omitir o combinar, en métodos consistentes con varias realizaciones de la presente divulgación, dentro del alcance de las reivindicaciones anexas.

10

REIVINDICACIONES

- 1.- Un método (1400) implementado en un decodificador, que comprende:
recibir (1401), por un receptor del decodificador, una corriente de bits que comprende datos codificados de una sub-imagen que contiene un grupo de mosaicos;
en donde la corriente de bits además comprende una cabecera de grupo de mosaicos asociada con el grupo de mosaicos,
obtener (1403), por el procesador, una dirección de grupo de mosaico, en donde la dirección de grupo de mosaicos incluye un índice de grupo de mosaico de nivel de sub-imagen del grupo de mosaicos cuando el grupo de mosaicos es un grupo de mosaico rectangular;
obtener (1403), por el procesador, un ID de identificador de la sub-imagen que contiene el grupo de mosaicos, en donde la dirección del grupo de mosaicos y el ID de sub-imagen se obtienen del encabezado del grupo de mosaicos;
posicionar (1407), por el procesador, el grupo de mosaicos de la sub-imagen basándose en la dirección de grupo de mosaicos y el ID de sub-imagen; y
decodificar (1409), por el procesador, el grupo de mosaicos de la sub-imagen basándose en la dirección del grupo de mosaicos.
2. El método de la reivindicación 1, en donde el grupo de mosaicos es un grupo de mosaicos rectangular cuando el `rect_tile_group_flag` es igual a uno.
3. El método de la reivindicación 1 o 2, en donde la dirección de grupo de mosaicos se obtiene de una variable `tile_grou_address` y en donde el ID de sub-imagen se obtiene de una variable `tile_group_subpic_id`.
- 4.- El método de una de las reivindicaciones 1-3, en donde la corriente de bits además comprende un conjunto de parámetros de secuencia (SPS), en donde el SPS contiene una descripción de la sub-imagen, y en donde el ID de sub-imagen hace referencia a la descripción de la sub-imagen.
- 5.- El método de la reivindicación 4, en donde la descripción de la sub-imagen incluye una anchura de la sub-imagen en unidades de bloques de árbol de codificación (CTB), una altura de la sub-imagen en unidades de CTBs, y una compensación de la sub-imagen especificada como una posición vertical de una unidad de árbol de codificación superior izquierda (CTU) de la sub-imagen y una posición horizontal de la CTU superior izquierda de la sub-imagen.
6. Un método (1300) implementado en un codificador, el método que comprende:
dividir (1301), mediante un procesador del codificador, una sub-imagen en un grupo de mosaicos;
determinar (1303), por el procesador, una dirección de grupo de mosaicos, en donde la dirección del grupo de mosaicos incluye un índice de grupo de mosaicos de nivel de sub-imagen del grupo de mosaicos cuando el grupo de mosaicos es un grupo de mosaicos rectangular;
determinar (1303), por el procesador, un identificador de sub-imagen, ID, de la sub-imagen que contiene el grupo de mosaicos;
codificar (1305), por el procesador, la dirección del grupo de mosaicos en una corriente de bits;
codificar (1305), por el procesador, la ID de la sub-imagen en la corriente de bits;
codificar (1307), por el procesador, el grupo de mosaicos de la sub-imagen en la corriente de bits;
en donde la corriente de bits comprende además un encabezado de grupo de mosaicos asociado con el grupo de mosaicos, en donde la dirección de grupo de mosaicos y el ID de sub-imagen están codificados en el encabezado de grupo de mosaicos.
7. El método de la reivindicación 6, en donde el grupo de mosaicos es un grupo de mosaicos rectangular cuando `rect_tile_group_flag` es igual a uno.
8. El método de la reivindicación 6 o 7, en donde la dirección de grupo de mosaicos está codificada en una variable de dirección_de_grupo de mosaicos, y en donde el ID de sub-imagen está codificado en una variable `group_subpic_id_variable`.
9. El método de una de las reivindicaciones 6-8, que comprende además codificar (1309), por el procesador, un conjunto de parámetros de secuencia, SPS, en la corriente de bits, en donde el SPS contiene una descripción de la sub-imagen y en donde el ID de sub-imagen hace referencia a la descripción de la sub-imagen.
10. El método de la reivindicación 9, en donde la descripción de la sub-imagen incluye un ancho de la sub-imagen en unidades de bloques de árbol de codificación, CTB, una altura de la sub-imagen en unidades de CTB y un desplazamiento de la sub-imagen especificado como una posición vertical de una unidad de árbol de codificación superior izquierda, CTU, de la sub-imagen y una posición horizontal de la CTU superior izquierda de la sub-imagen.

11. Un dispositivo de codificación de vídeo (1200) que comprende:

un procesador (1230), un receptor (1210) acoplado al procesador (1230), una memoria (1232) acoplada al procesador (1230), y un transmisor (1210) acoplado al procesador (1230), en donde el procesador (1230), receptor (1210), memoria (1232) y transmisor (1210) están configurados para realizar el método de una de las reivindicaciones 1 a 10.

12. Un medio no transitorio legible por computadora que comprende un producto de programa informático para su uso mediante un dispositivo de codificación de vídeo, el producto de programa informático que comprende instrucciones ejecutables por computadora almacenadas en el medio no transitorio legible por computadora tal que cuando son ejecutadas por un procesador hacen que el dispositivo de codificación de vídeo realice el método de una de las reivindicaciones 1-10.

13. Un producto de programa informático que comprende un código de programa para realizar el método de acuerdo con una de las reivindicaciones 1 a 10 cuando se ejecuta en una computadora o un procesador.

14. Una corriente de bits de vídeo codificado, en donde la corriente de bits de vídeo codificado comprende datos codificados de una sub-imagen que contiene un grupo de mosaicos, y un encabezado de grupo de mosaicos asociado con el grupo de mosaicos, en donde el encabezado de grupo de mosaicos comprende una dirección de grupo de mosaicos, en donde la dirección de grupo de mosaicos incluye un índice de grupo de mosaicos de nivel de sub-imagen del grupo de mosaicos cuando el grupo de mosaicos es un grupo de mosaicos rectangular; en donde el encabezado de grupo de mosaicos comprende además un ID de identificador de la sub-imagen que contiene el grupo de mosaicos.

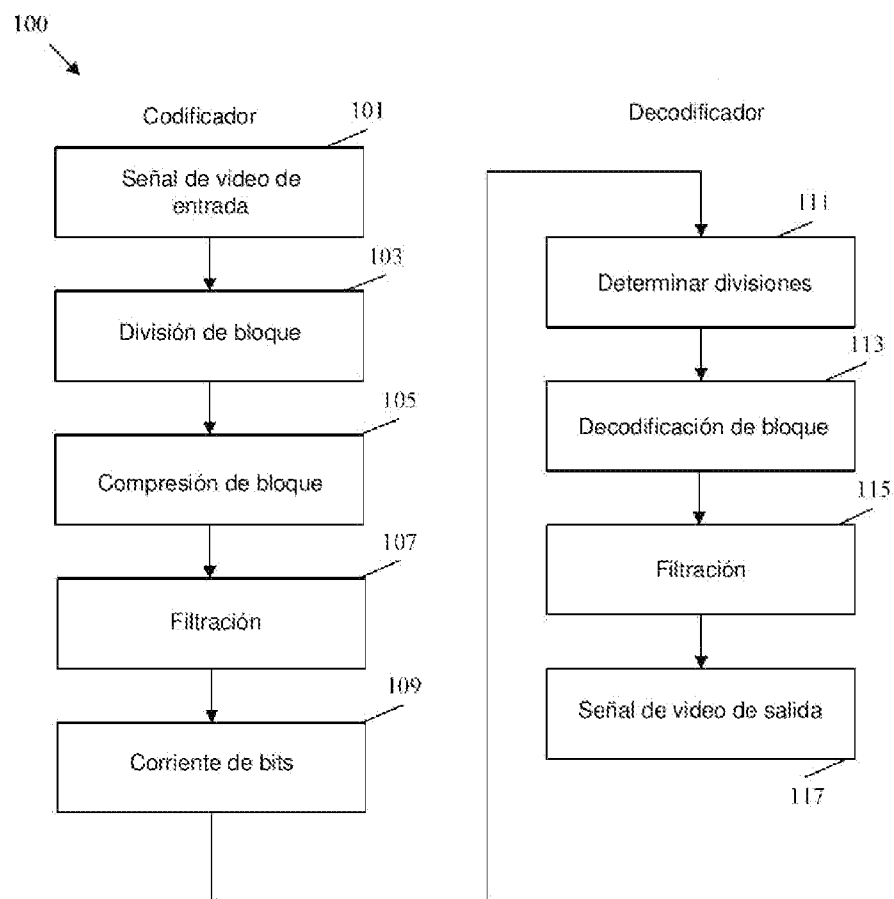


FIGURA 1

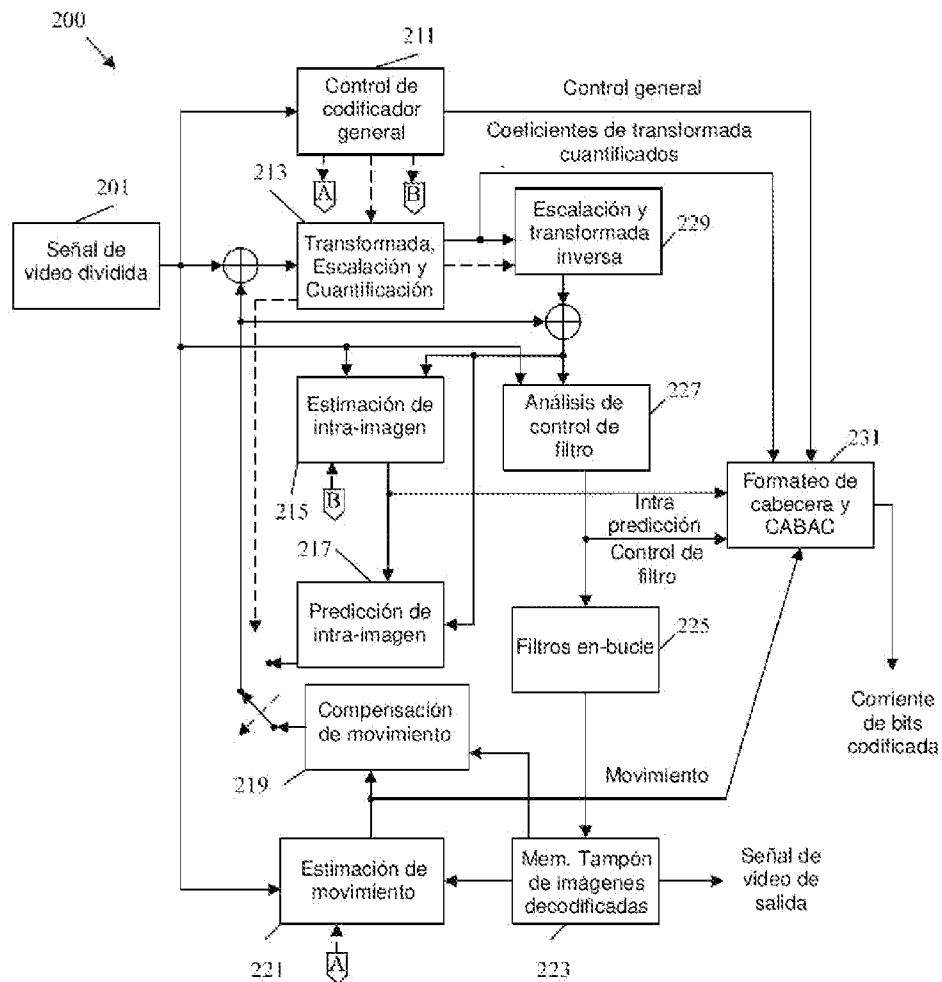


FIGURA 2

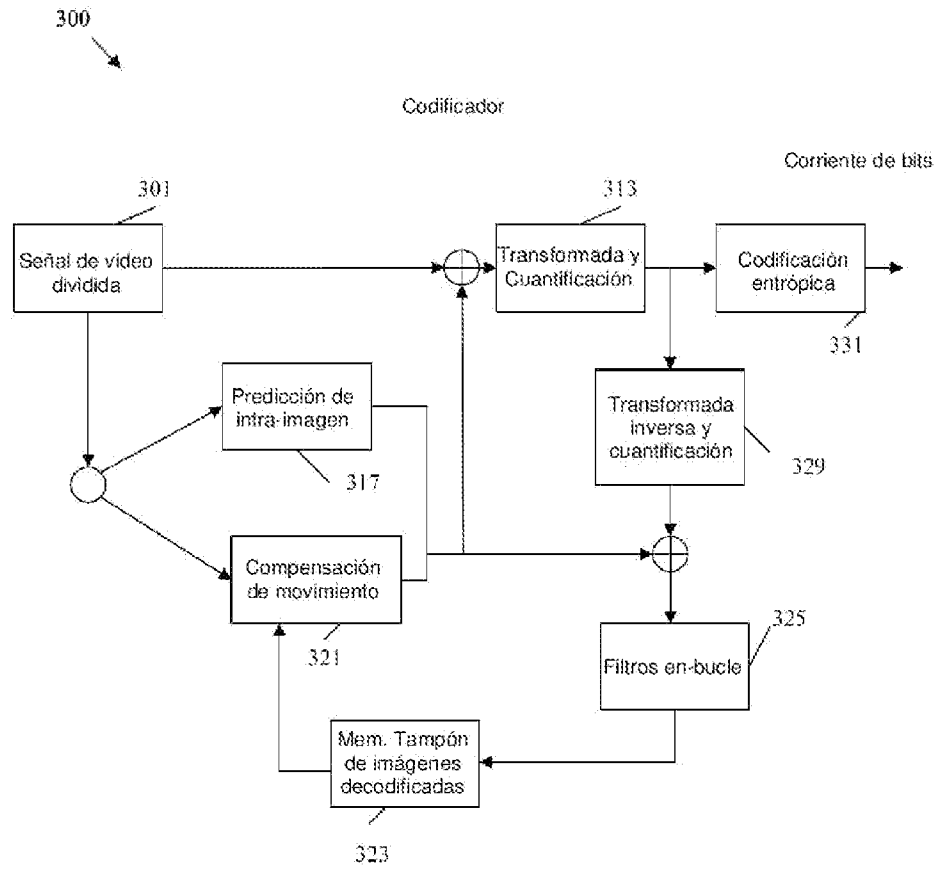


FIGURA 3

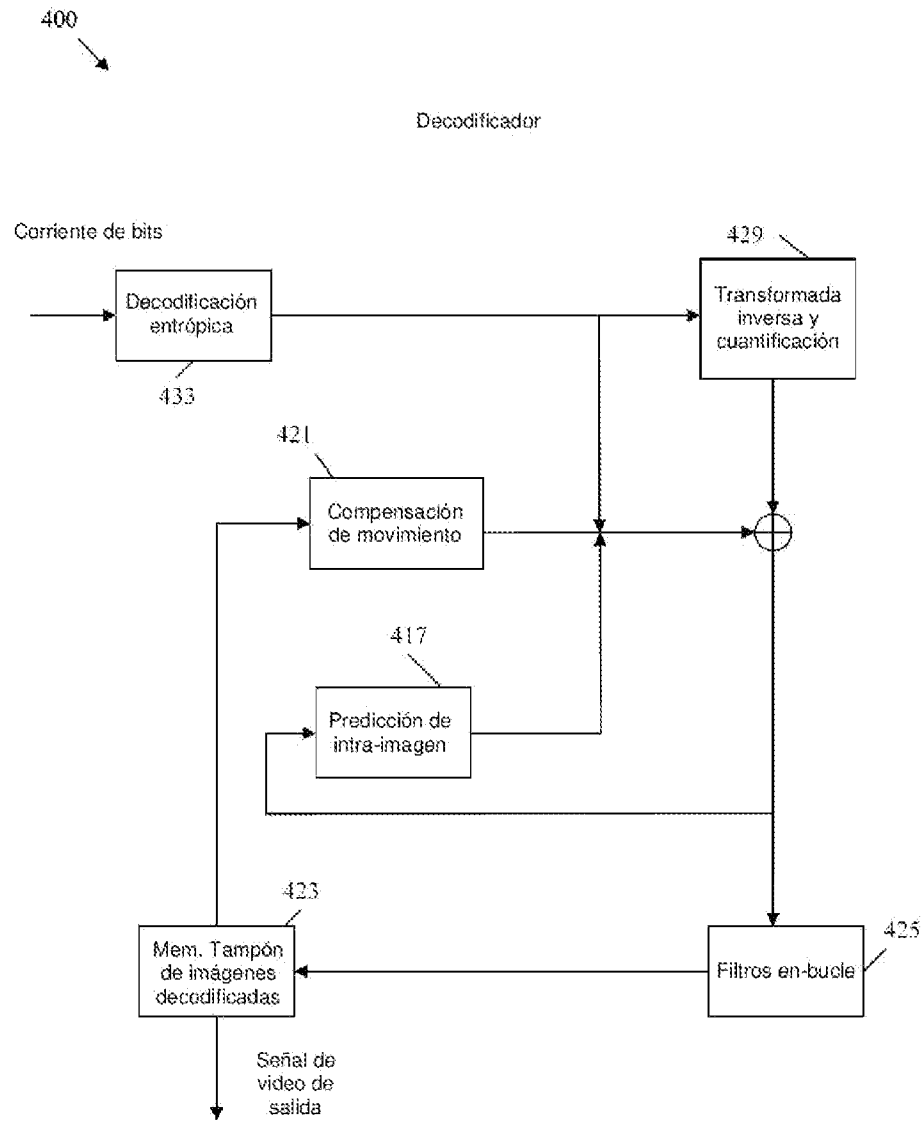


FIGURA 4

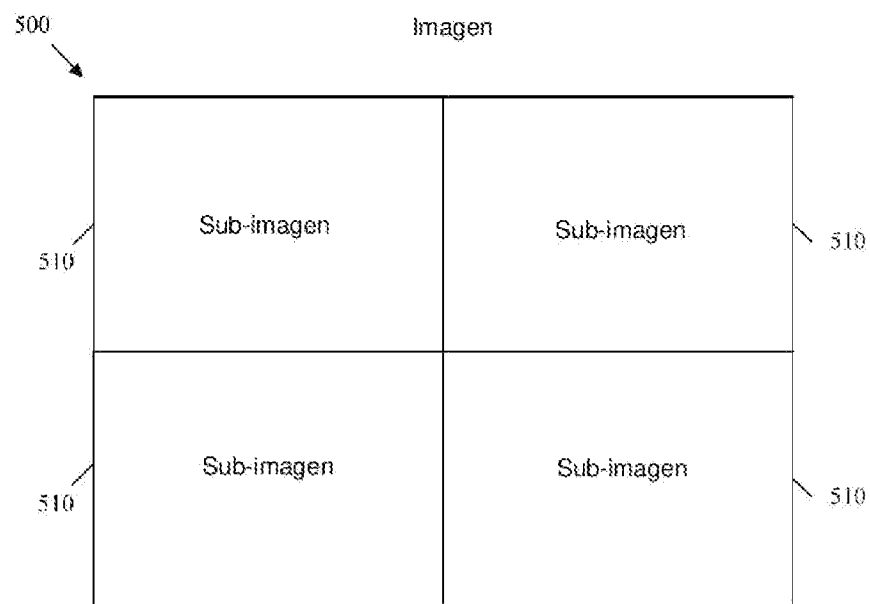


FIGURA 5A

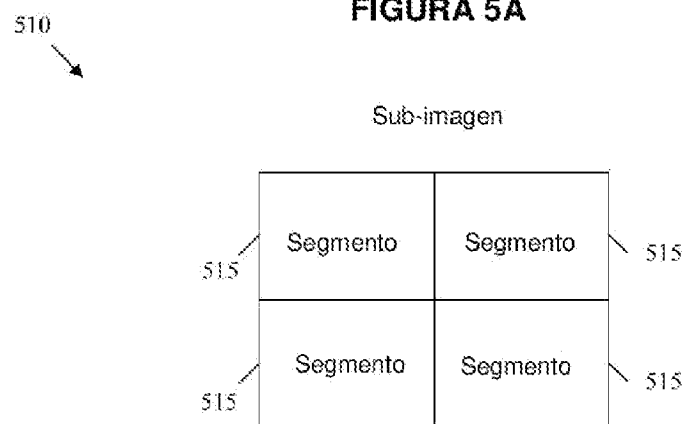


FIGURA 5B

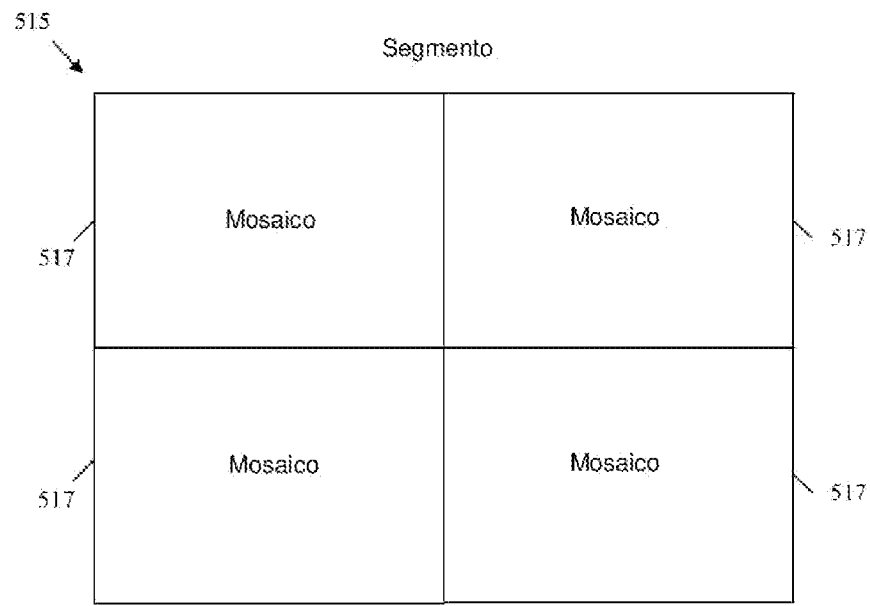


FIGURA 5C

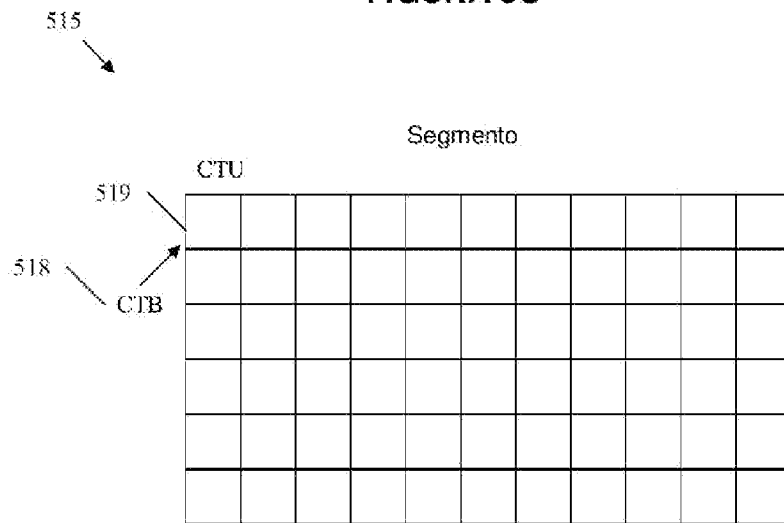


FIGURA 5D

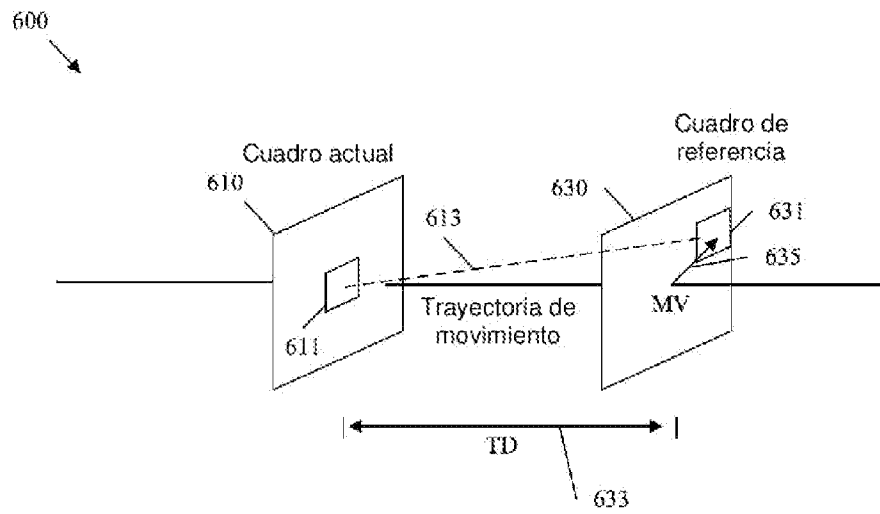


FIGURA 6

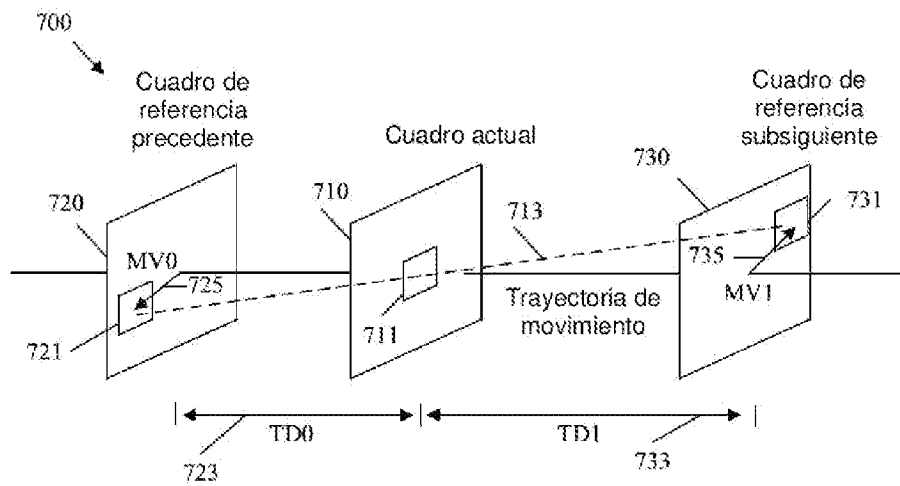


FIGURA 7

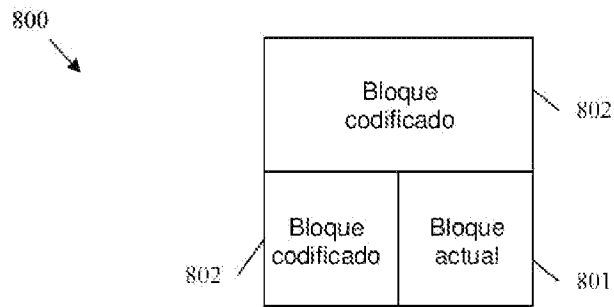


FIGURA 8

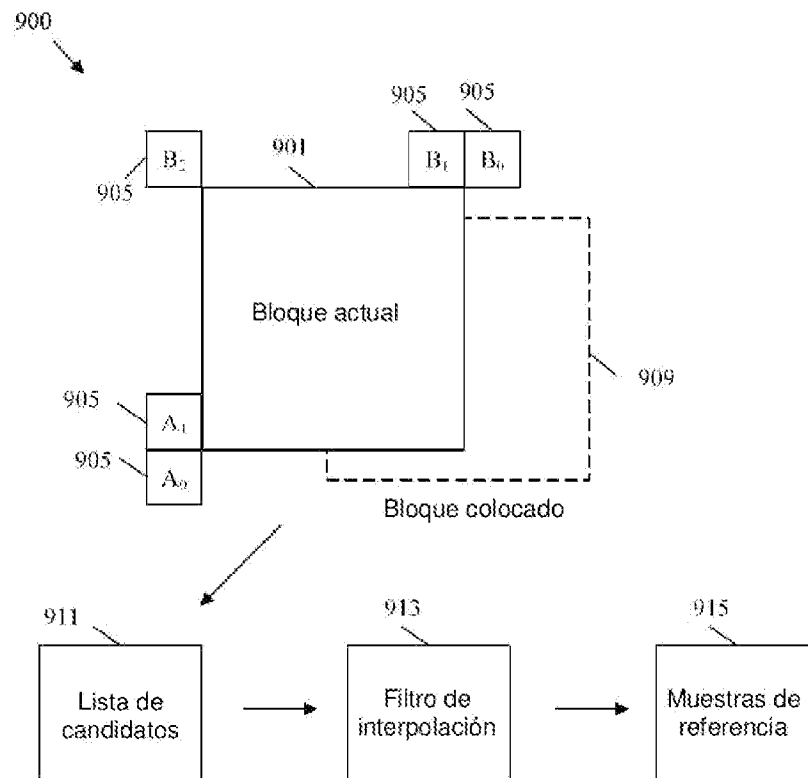


FIGURA 9

1000
↓

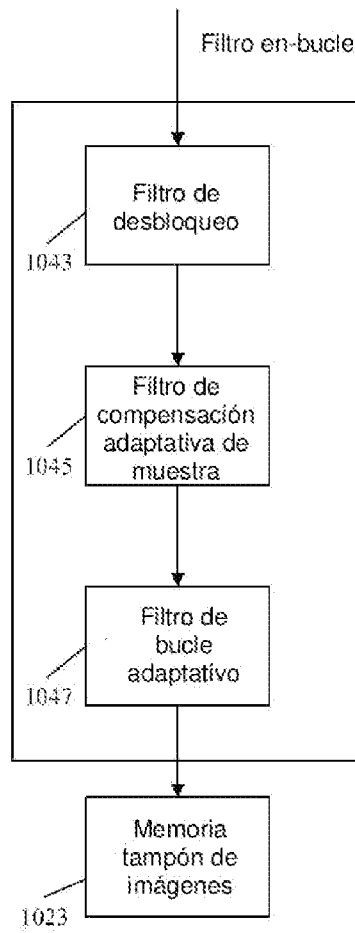


FIGURA 10

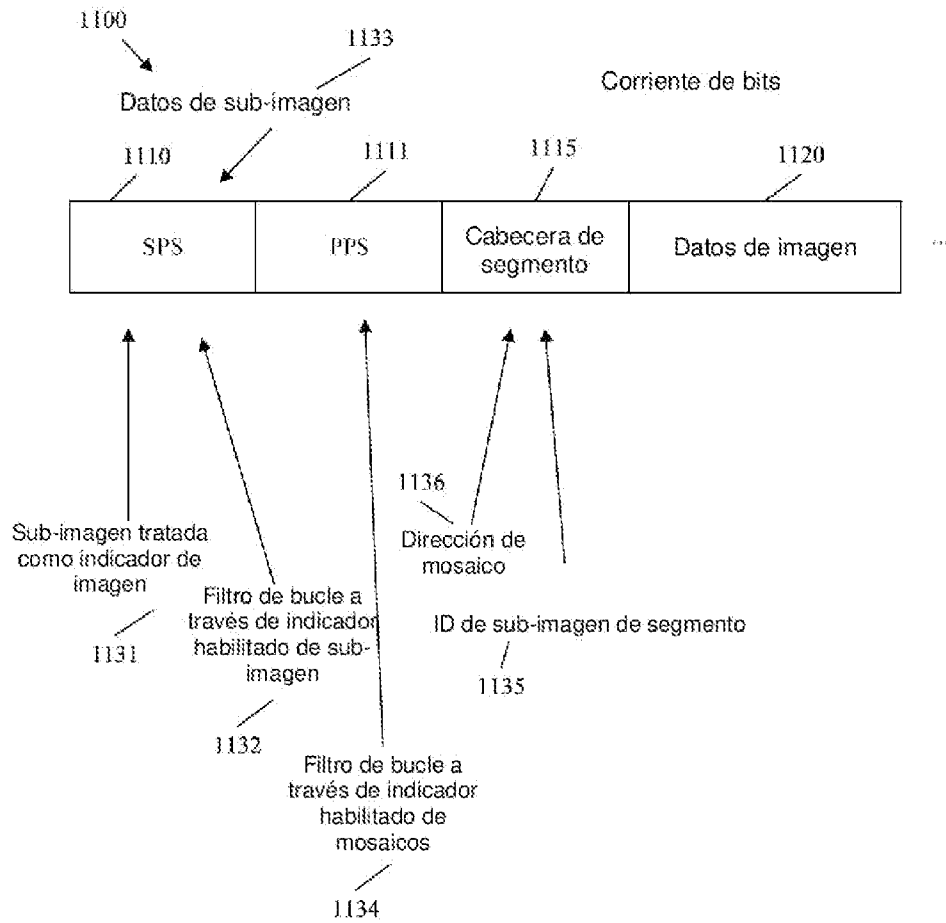


FIGURA 11

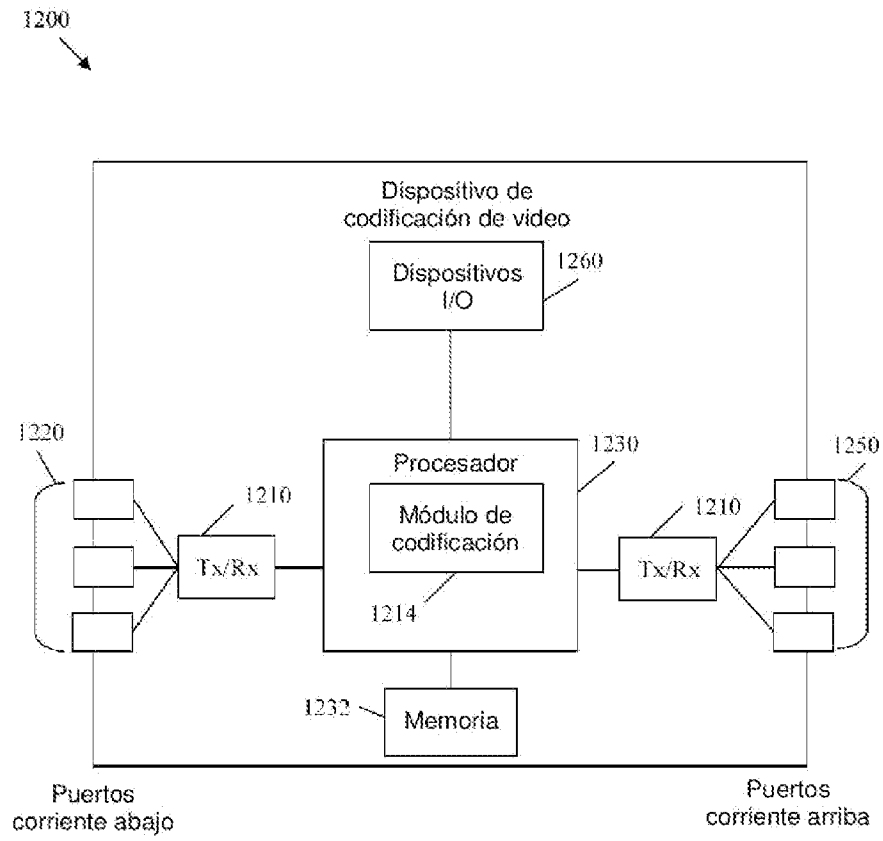


FIGURA 12

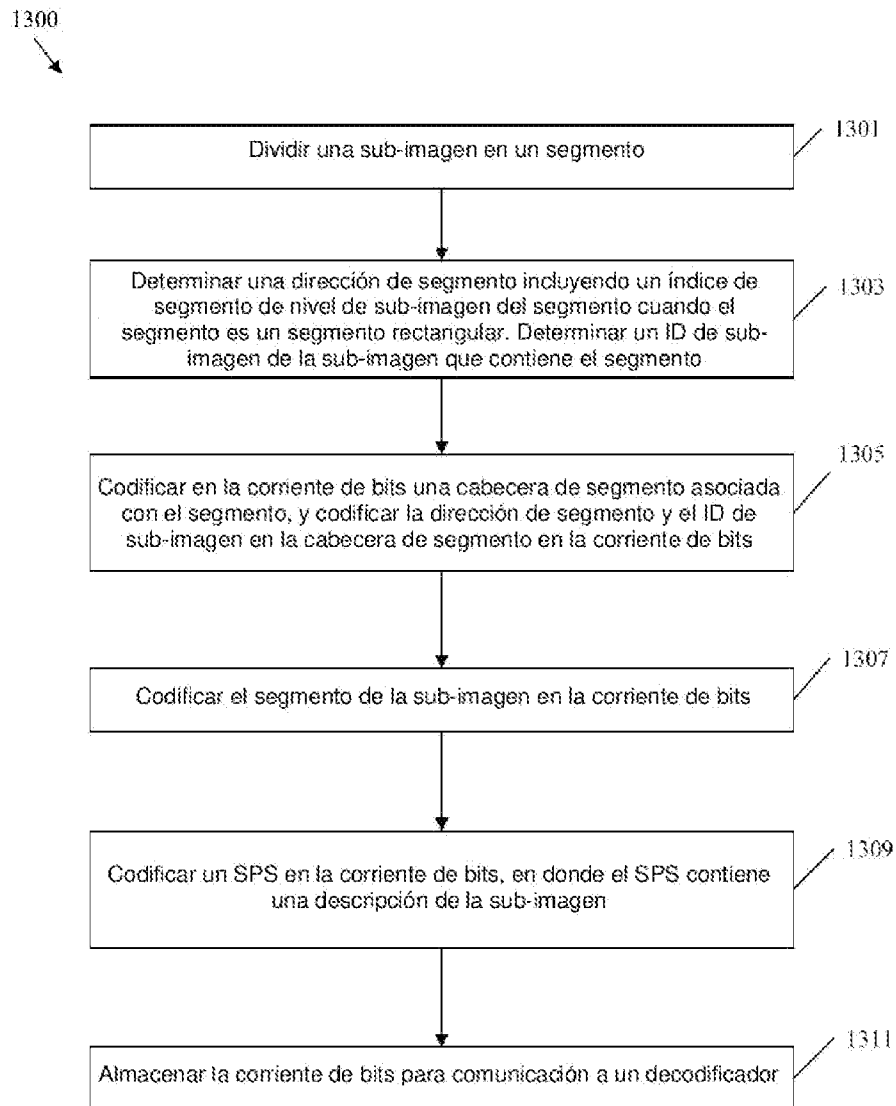


FIGURA 13

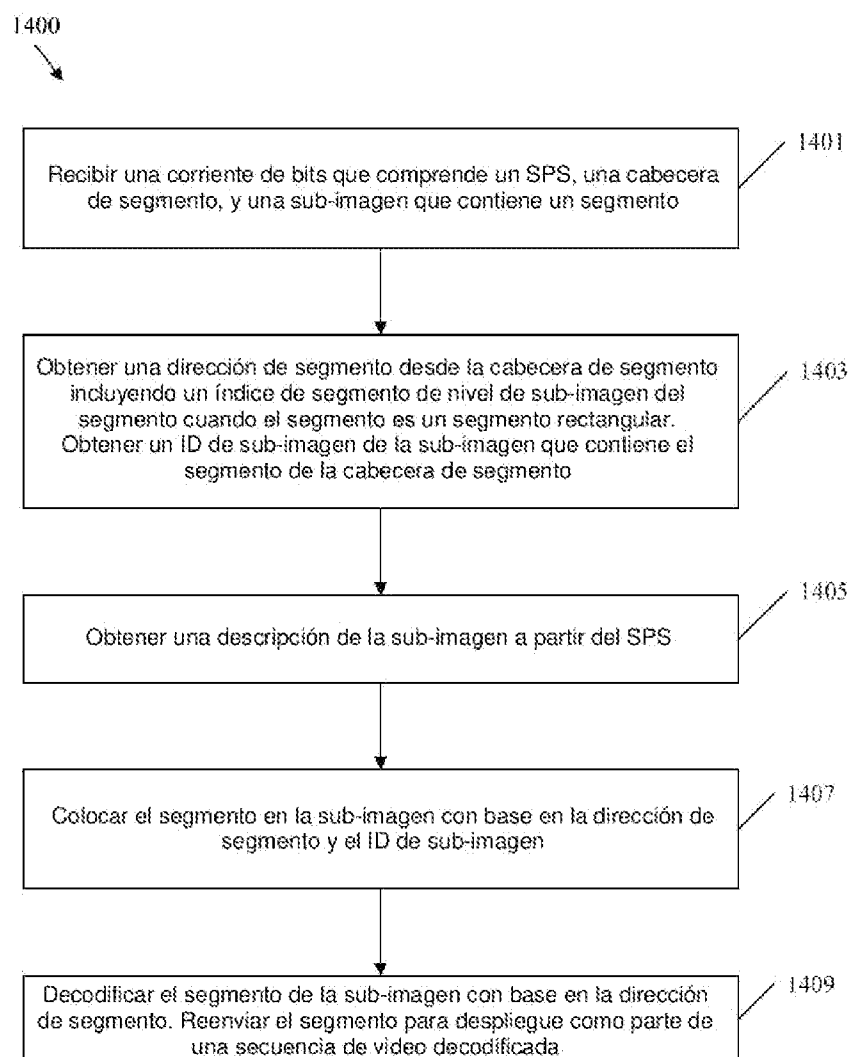


FIGURA 14

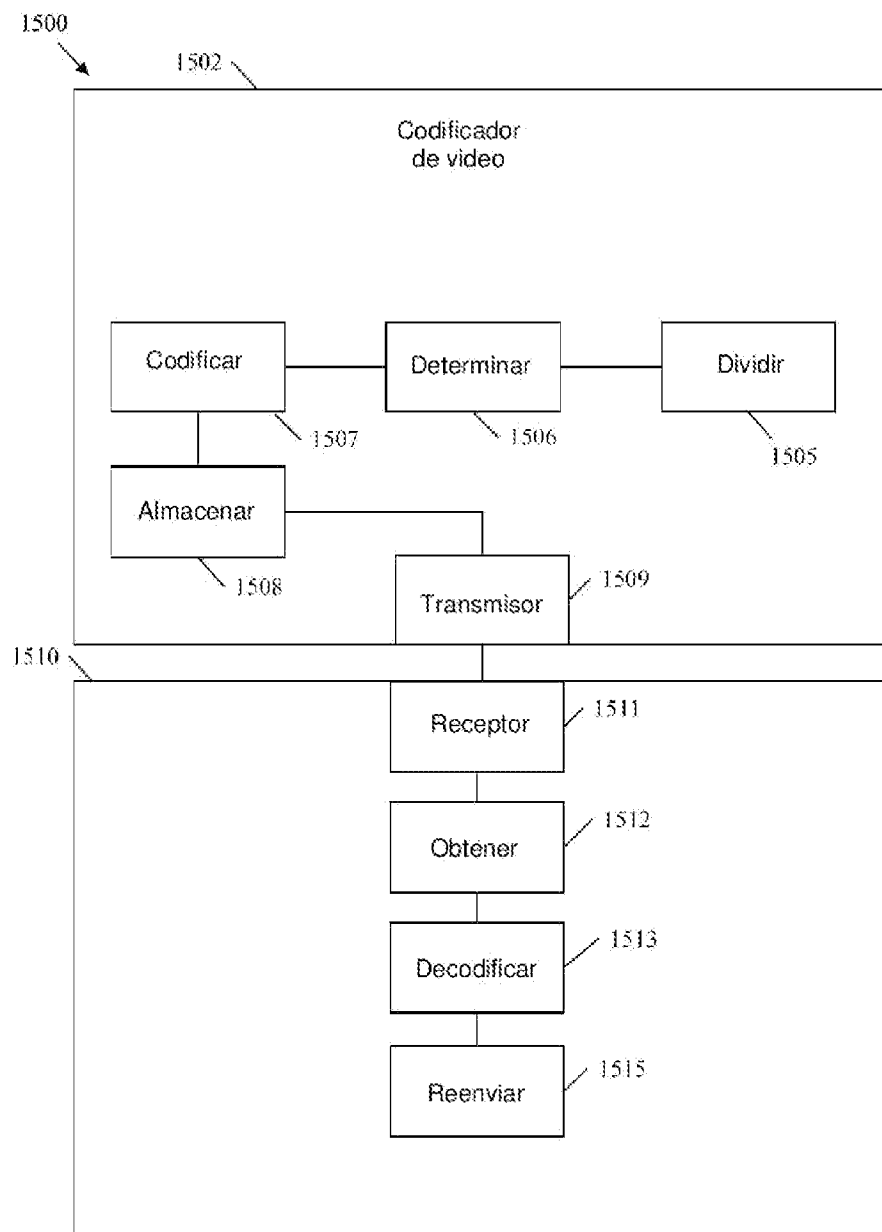


FIGURA 15