

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 004 733**

51 Int. Cl.:

H04B 1/66 (2006.01)

H04N 19/70 (2014.01)

H04N 19/117 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.02.2020** **PCT/US2020/019920**

87 Fecha y número de publicación internacional: **03.09.2020** **WO20176635**

96 Fecha de presentación y número de la solicitud europea: **26.02.2020** **E 20763133 (4)**

97 Fecha y número de publicación de la concesión europea: **27.11.2024** **EP 3931747**

54 Título: **Espacios de valor de identificador de conjunto de parámetros de adaptación en codificación de vídeo**

30 Prioridad:

27.02.2019 US 201962811358 P

11.03.2019 US 201962816753 P

21.05.2019 US 201962850973 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
13.03.2025

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.00%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN

72 Inventor/es:

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

74 Agente/Representante:

ELZABURU, S.L.P

ES 3 004 733 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

Espacios de valor de identificador de conjunto de parámetros de adaptación en codificación de vídeo

5 CAMPO TÉCNICO

La presente invención está relacionada generalmente con la codificación de vídeo y específicamente está relacionada con señalización eficiente de parámetros de herramientas de codificación utilizados para comprimir datos de vídeo en codificación de vídeo.

10

ANTECEDENTES

La cantidad de datos de vídeo necesarios para representar incluso un vídeo relativamente corto pueden ser sustanciales, lo que puede tener como resultado dificultades cuando los datos se van a transmitir o comunicar de otro modo por una red de comunicaciones con capacidad limitada de ancho de banda. Así, los datos de vídeo generalmente se comprimen antes de ser comunicados por redes de telecomunicaciones diurnas por módem. El tamaño de un vídeo también podría ser un asunto cuando el vídeo se almacena en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Los dispositivos de compresión de vídeo a menudo usan software y/o hardware en la fuente para codificar los datos de vídeo antes de la transmisión o almacenamiento, disminuyendo de ese modo la cantidad de datos necesarios para representar imágenes de vídeo digital. Los datos comprimidos entonces son recibidos en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de red limitados y demandas siempre crecientes de calidad de vídeo más alta, son deseables técnicas de compresión y descompresión mejoradas que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de imagen.

25

Y-K WANG (HUAINEI) ET AL: "AHG17: On header parameter set (HPS)", 125. MPEG MEETING; 20190114-20190118; MARRUECOS; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), n.º m45394, 16 de enero de 2019 (2019-01-16), divulga el conjunto de parámetros de adaptación de diseño (APS) para carro de parámetros de ALF

30

HENDRY ET AL: "Aps partial update APS update with one or multiple references", 100. MPEG MEETING; 30-4-2012 4-5-2012; GÉNOVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), n.º m24320, 16 de abril de 2012 (2012-04-16), divulga una actualización de APS con una o varias referencias.

35 COMPENDIO

En una realización, la invención incluye un método implementado en un decodificador, el método comprende: recibir, por un receptor del decodificador, un flujo de bits que comprende sintaxis de datos de vídeo, en donde los datos de vídeo comprenden una pluralidad de imágenes; particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos codificados, en donde la sintaxis comprende una pluralidad de conjuntos de parámetros de adaptación (APS) que incluye una pluralidad de tipos de APS asociados con un segmento codificado de los segmentos codificados, en donde cada APS incluye un identificador de APS (ID), y en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS; decodificar, por el procesador, el segmento codificado usa parámetros de la pluralidad de APS obtenidos en función de los ID de APS; en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, y un tipo de correlación de luma con escalado de croma, LMCS, que contiene parámetros de LMCS; en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalos valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS; en donde cada APS incluye un tipo de parámetro de APS, *aps_params_type*, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS. Un APS se usa para mantener datos que están relacionados con múltiples segmentos sobre múltiples imágenes. La presente invención describe diversas mejoras relacionadas con APS. En la presente ejemplo, cada APS incluye un ID de APS. Además, cada tipo de APS incluye un espacio de valores separado para correspondientes ID de APS. Tales espacios de valores pueden superponerse. Por consiguiente, un APS de un primer tipo, tal como un APS de filtro de bucle adaptativo (ALF), puede incluir el mismo ID de APS como APS de un segundo tipo, este tipo de correlación luma con APS de escalado croma (LMCS). Esto se consigue al identificar cada APS por una combinación de un tipo de parámetro de APS y un ID de APS. Al permitir que cada tipo de APS incluya un espacio de valores diferente, el códec no necesita comprobar conflictos de ID en tipos de APS. Además, al permitir que los espacios de valor se superpongan, el códec puede evitar emplear valores de ID más grandes, lo que tiene como resultado ahorros de bits. Como tal, emplear espacios de valor superpuestos separados para los APS de diferentes tipos aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador.

65

Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde los espacios de valores separados se superponen.

5 Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde la pluralidad de tipos de APS incluye un tipo de lista de escalado que contiene parámetros de lista de escalado, en donde los parámetros de lista de escalado se incluyen en una lista de escalado para un proceso de transformada.

10 Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde un APS actual incluye un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual, en donde el ID de APS actual se relaciona con un ID de APS anterior asociado con un APS anterior de un mismo tipo que el APS actual, y en donde el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual.

15 Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde cada APS se identifica por una combinación de un tipo de APS actual y un ID de APS actual.

En una realización, la invención incluye un método implementado en un codificador, el método comprende: recibir datos de vídeo que comprenden una pluralidad de imágenes; particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos; determinar, por el procesador, una pluralidad de tipos de parámetros empleados para codificar un segmento de la pluralidad de segmentos; codificar, por un procesador del codificador, el segmento hasta un flujo de bits como segmento codificado; codificar en el flujo de bits, por el procesador, la pluralidad de tipos de parámetros en una pluralidad de APS al incluir la pluralidad de tipos de parámetros en una pluralidad de tipos de APS; en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, y un tipo de correlación de luma con escalado de croma, LMCS, que contiene parámetros de LMCS; en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalos valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS; asignar, por el procesador, un ID de APS a cada uno de los APS de manera que cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS; codificar, por el procesador, cada ID de APS en la pluralidad de APS; en donde cada APS incluye un tipo de parámetro de APS, `aps_params_type`, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS. Un APS se usa para mantener datos que están relacionados con múltiples segmentos sobre múltiples imágenes. La presente invención describe diversas mejoras relacionadas con APS. En la presente ejemplo, cada APS incluye un ID de APS. Además, cada tipo de APS incluye un espacio de valores separado para correspondientes ID de APS. Tales espacios de valores pueden superponerse. Por consiguiente, un APS de un primer tipo (p. ej. un APS de ALF) puede incluir el mismo ID de APS como APS de un segundo tipo (p. ej. un APS de LMCS). Esto se consigue al identificar cada APS por una combinación de un tipo de parámetro de APS y un ID de APS. Al permitir que cada tipo de APS incluya un espacio de valores diferente, el códec no necesita comprobar conflictos de ID en tipos de APS. Además, al permitir que los espacios de valor se superpongan, el códec puede evitar emplear valores de ID más grandes, lo que tiene como resultado ahorros de bits. Como tal, emplear espacios de valor superpuestos separados para los APS de diferentes tipos aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador.

Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde los espacios de valores separados se superponen.

50 Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde la pluralidad de tipos de APS incluye un tipo de lista de escalado que contiene parámetros de lista de escalado, en donde los parámetros de lista de escalado se incluyen en una lista de escalado para un proceso de transformada.

55 Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde un APS actual incluye un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual, en donde el ID de APS actual se relaciona con un ID de APS anterior asociado con un APS anterior de un mismo tipo que el APS actual, y en donde el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual.

Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde cada APS se identifica por una combinación de un tipo de APS actual y un ID de APS actual.

65 En una realización, la invención incluye un dispositivo de codificado de vídeo que comprende: un procesador, un receptor acoplado al procesador, una memoria acoplada al procesador, y un transmisor acoplado al procesador,

en donde procesador, receptor, memoria y transmisor se configuran para realizar el método de cualquiera de los aspectos anteriores

En una realización, la invención incluye un medio no transitorio legible por ordenador que comprende un producto de programa informático para su uso por un dispositivo de codificado de vídeo, el producto de programa informático comprende instrucciones ejecutables por ordenador almacenadas en el medio no transitorio legible por ordenador de tal manera que cuando es ejecutado por un procesador hace que el dispositivo de codificado de vídeo realice el método de cualquiera de cualquiera de los aspectos anteriores.

En una realización, la invención incluye un decodificador que comprende: una unidad de recepción, configurada para recibir un flujo de bits que comprende sintaxis de datos de vídeo, en donde los datos de vídeo comprenden una pluralidad de imágenes; una unidad de partición, configurada para particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos codificados, en donde la sintaxis comprende una pluralidad de conjuntos de parámetros de adaptación, APS, que incluye una pluralidad de tipos de APS asociados con un segmento codificado de los segmentos codificados, en donde cada APS incluye un identificador de APS, ID, y en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS; y una unidad de decodificación, configurada para decodificar el segmento codificado usando parámetros de la pluralidad de APS obtenidos en función de los ID de APS; en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, un tipo de correlación de luma con escalado de croma, LMCS, que contiene parámetros de LMCS; en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalos valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS; en donde cada APS incluye un tipo de parámetro de APS, `aps_params_type`, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS.

Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde el decodificador se configura además para realizar el método de cualquiera de los aspectos anteriores

En una realización, la invención incluye un codificador que comprende: una unidad de recepción, configurada para recibir datos de vídeo que comprenden una pluralidad de imágenes; una unidad de partición, configurada para particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos; una unidad de determinación, configurada para determinar una pluralidad de tipos de parámetros empleados para codificar un segmento; una unidad de codificación, configurada para codificar el segmento en un flujo de bits como segmento codificado; codificar la pluralidad de tipos de parámetros en una pluralidad de conjuntos de parámetros de adaptación, APS, del flujo de bits al incluir la pluralidad de tipos de parámetros en una pluralidad de tipos de APS; y codificar cada identificador de APS, ID, en la pluralidad de APS; y una unidad de asignación, configurada para asignar un ID de APS a cada uno de los APS, en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS; en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, y un tipo de correlación de luma con escalado de croma, LMCS, que contiene parámetros de LMCS; en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalos valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS; en donde cada APS incluye un tipo de parámetro de APS, `aps_params_type`, set a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS

Opcionalmente, en cualquiera de los aspectos anteriores, se proporciona otra implementación del aspecto, en donde el codificador se configura además para realizar el método de cualquiera de los aspectos anteriores

Con el propósito de claridad, cualquiera de la realizaciones anteriores puede combinarse con una cualquiera o más de las otras realizaciones anteriores para crear un nueva realización dentro del alcance de la presente invención.

Estos y otros rasgos se entenderán más claramente a partir de la siguiente descripción detallada tomada conjuntamente con los dibujos adjuntos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Para un entendimiento más completo de la esta invención, ahora se hace referencia a la siguiente breve descripción, tomada en relación a los dibujos adjuntos y la descripción detallada, en donde numerales de referencia semejantes representan partes semejantes.

La FIGURA 1 es un diagrama de flujo de un método de ejemplo de codificar una señal de vídeo.

La FIGURA 2 es un diagrama esquemático de un ejemplo de sistema de codificación y decodificación (códec) para codificación de vídeo.

La FIGURA 3 es un diagrama esquemático que ilustra un codificador de vídeo de ejemplo.

La FIGURA 4 es un diagrama esquemático que ilustra un decodificador de vídeo de ejemplo.

La FIGURA 5 es un diagrama esquemático que ilustra un flujo de bits de ejemplo que contiene múltiples tipos de conjunto de parámetros de adaptación (APS) que incluye tipos diferentes de parámetros de herramienta de codificación.

La FIGURA 6 es un diagrama esquemático que ilustra un mecanismo de ejemplo para asignar identificadores de APS (ID) a tipos diferentes de APS en diferentes espacios de valores.

La FIGURA 7 es un diagrama esquemático de un dispositivo de codificado de vídeo de ejemplo.

La FIGURA 8 es un diagrama de flujo de un método de ejemplo de codificar una secuencia de vídeo en el flujo de bits empleando una pluralidad de tipos de APS.

La FIGURA 9 es un diagrama de flujo de un método de ejemplo de decodificar una secuencia de vídeo a partir de un flujo de bits empleando una pluralidad de tipos de APS.

La FIGURA 10 es un diagrama esquemático de un sistema de ejemplo para codificar una secuencia de vídeo de imágenes hasta un flujo de bits, empleando una pluralidad de tipos de APS.

DESCRIPCIÓN DETALLADA

Se debe entender al comienzo que, aunque a continuación se proporciona una implementación ilustrativa de una o más realizaciones, los sistemas y/o métodos divulgados pueden implementarse usando cualquier número de técnicas, ya sean conocidas o existentes actualmente. La invención de ninguna manera debe limitarse a las implementaciones ilustrativas, los dibujos y las técnicas ilustrados a continuación, incluidos los diseños e implementaciones ejemplares ilustrados y descritos en esta memoria, sino que puede modificarse dentro del alcance de las reivindicaciones anexas junto con su alcance completo de equivalentes.

En esta memoria se emplean diversos acrónimos, filtro de bucle adaptativo (ALF), conjunto de parámetros de adaptación (APS), bloque de árbol de codificación (CTB), unidad de árbol de codificación (CTU), unidad de codificación (CU), secuencia de vídeo codificado (CVS), retransmisión adaptativa dinámica por protocolo de transferencia de hipertexto (DASH), punto de acceso intraaleatorio (IRAP), equipo conjunto de expertos en vídeo (JVET), conjunto de mosaicos con restricción de movimiento (MCTS), unidad de transferencia máxima (MTU), capa de abstracción de red (NAL), recuento de orden de imágenes (POC), carga útil de secuencia de bytes sin procesar (RBSP), conjunto de parámetros de secuencia (SPS), codificación de vídeo versátil (VVC) y borrador de trabajo (WD).

Se pueden emplear muchas técnicas de compresión de vídeo para reducir el tamaño de los archivos de vídeo con una pérdida mínima de datos. Por ejemplo, las técnicas de compresión de vídeo pueden incluir la realización de una predicción espacial (por ejemplo, intraimagen) y/o temporal (por ejemplo, interimágenes) para reducir o eliminar la redundancia de datos en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una imagen de vídeo o una parte de una imagen de vídeo) puede partitionarse en bloques de vídeo, que también pueden denominarse bloques de árbol, bloques de árbol de codificación (CTB), unidades de árbol de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un segmento intracodificado (I) de una imagen se codifican utilizando la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento de predicción unidireccional (P) o bidireccional (B) intercodificado de una imagen pueden codificarse empleando la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen o la predicción temporal con respecto a las muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse fotogramas y/o imágenes, y las imágenes de referencia pueden denominarse fotogramas de referencia y/o imágenes de referencia. La predicción espacial o temporal da como resultado un bloque predictivo que representa un bloque de imagen. Los datos residuales representan las diferencias de píxeles entre el bloque de imagen original y el bloque predictivo. Por consiguiente, un bloque intercodificado se codifica según un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica según un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales se pueden transformar del dominio de píxeles a un dominio de transformada. Estos dan como resultado coeficientes de transformada residuales, que pueden cuantizarse. Los coeficientes de transformada cuantizados pueden disponerse inicialmente en una distribución bidimensional. Los coeficientes de transformada cuantizados se pueden escanear para producir un vector unidimensional de coeficientes de transformada. Se puede aplicar codificación por entropía para lograr una compresión aún mayor. Dichas técnicas de compresión de vídeo se describen con mayor detalle a continuación.

Para garantizar que un vídeo codificado se pueda decodificar con precisión, el vídeo se codifica y decodifica según los estándares de codificación de vídeo correspondientes. Los estándares de codificación de vídeo incluyen el Sector de Normalización de la Unión Internacional de Telecomunicaciones (UIT) (UIT-T) H.261, Grupo de Expertos en Cinematografía (MPEG) -1 Parte 2 de la Organización Internacional de Normalización / Comisión Electrotécnica

Internacional (ISO/IEC), UIT-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 conocida como ITU-T H.265 o MPEG-H Parte 2. AVC incluye extensiones como Codificación de vídeo escalable (SVC), Codificación de vídeo de vista múltiple (MVC) y Codificación de vídeo de vista múltiple más profundidad (MVC+D), y AVC tridimensional (3D) (3D-AVC). HEVC incluye extensiones como HEVC escalable (SHVC), HEVC de vista múltiple (MV-HEVC) y HEVC 3D (3D-HEVC). El equipo conjunto de expertos en vídeo (JVET) de ITU-T e ISO/IEC ha comenzado a desarrollar un estándar de codificación de vídeo denominado Codificación de vídeo versátil (VVC). VVC se incluye en un Borrador de Trabajo (WD), que incluye JVET-M1001-v5 y JVET-M1002-v1, que proporciona una descripción de algoritmo, una descripción de lado codificador de la VVC WD, y software de referencia.

Se codifican secuencias de vídeo empleando diversas herramientas de codificación. El codificador selecciona parámetros para la herramientas de codificación con el objetivo de aumentar la compresión con una mínima pérdida de calidad cuando se decodifica la secuencia de vídeo. Las herramientas de codificación pueden ser relevantes para diferentes partes del vídeo en diferentes alcances. Por ejemplo, algunas herramientas de codificación son relevantes en un nivel de secuencia de vídeo, algunas herramientas de codificación son relevantes en un nivel de imagen, algunas herramientas de codificación son relevantes en un nivel de segmento, etc. Se puede emplear un APS para señalar información que se puede compartir por múltiples imágenes y/o múltiples segmentos en diferentes imágenes. Específicamente, un APS puede llevar parámetros de filtro de bucle adaptativo (ALF). La información de ALF puede no ser adecuada para señalar en el nivel de secuencia en un conjunto de parámetros de secuencia (SPS), en nivel de imagen en un conjunto de parámetros de imagen (PPS) o un encabezado de imagen, o en un nivel de segmento en un grupo de mosaicos/encabezado de segmento por diversas razones.

Si se señala información de ALF en el SPS, un codificador tiene para generar un nuevo SPS y una nueva imagen de IRAP cuando cambia la información de ALF. Las imágenes de IRAP reducen significativamente la eficiencia de codificación. Por lo que colocar información de ALF en un SPS es particularmente problemático para entornos de aplicación de retraso bajo que no emplea imágenes de IRAP frecuentes. La inclusión de información de ALF en el SPS también pueden inhabilitar la transmisión de fuera de banda de SPS. Transmisión de fuera de banda se refiere a transmisión de datos correspondientes en diferentes flujos de datos de transporte que el flujo de bits de vídeo (p. ej., en una descripción de muestra o anotación de muestra de un archivo multimedia, en un archivo de Protocolo de Descripción de Sesión (SDP), etc.). Señalizar información de ALF en un PPS también pueden ser problemático por razones similares. Específicamente, la inclusión de información de ALF en la PPS puede inhabilitar la transmisión de fuera de banda de PPS. Señalizar información de ALF en un encabezado de imagen también puede ser problemático. En algunos casos pueden no emplearse encabezados de imagen. Además, alguna información de ALF puede aplicarse a múltiples imágenes. Así señalar información de ALF en el encabezado de imagen provoca transmisión de información redundante, y por tanto desperdicia ancho de banda. Señalizar información de ALF en el grupo de mosaicos/encabezado de segmento también es problemático, puesto que la información de ALF puede aplicarse a múltiples imágenes y por tanto a múltiples grupos de mosaicos/segmentos. Por consiguiente, señalar información de ALF en el encabezado de grupo de mosaicos/segmento provoca transmisión de información redundante, y por tanto desperdicia ancho de banda.

En función de lo anterior, se puede emplear un APS para señalar parámetros de ALF. Sin embargo, sistemas de codificación de vídeo pueden emplear el APS exclusivamente para señalar parámetros de ALF. Un ejemplo de sintaxis y semántica de APS son de la siguiente manera:

adaptation_parameter_set_rbsp() {	Descriptor
adaptation_parameter_set_id	u(5)
alf_data()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

El adaptation_parameter_set_id proporciona un identificador para el APS para referencia por otros elementos de sintaxis. Los APS se pueden compartir por imágenes y pueden ser diferentes en diferentes grupos de mosaicos dentro de una imagen. El aps_extension_flag se establece igual a cero para especificar que no hay elementos de sintaxis aps_extension_data_flag presentes en la estructura de sintaxis de RBSP de APS. El aps_extension_flag

se establece igual a uno para especificar que hay elementos de sintaxis `aps_extension_data_flag` presentes en la estructura de sintaxis de RBSP de APS. El `aps_extension_data_flag` puede tener cualquier valor. La presencia y el valor de `aps_extension_data_flag` puede no afectar a la conformidad de decodificador a perfiles como se especifica en VVC. Decodificadores que se conforman a VVC pueden ignorar todos los elementos de sintaxis de `aps_extension_data_flag`.

Un ejemplo de sintaxis de encabezado de grupo de mosaicos relacionado con parámetros de ALF es de la siguiente manera:

tile_group_header() {	Descriptor
...	
if(sps_alf_enabled_flag) {	
tile_group_alf_enabled_flag	u(1)
if(tile_group_alf_enabled_flag)	
tile_group_aps_id	u(5)
}	
...	
}	

El `tile_group_alf_enabled_flag` se establece igual a uno para especificar que el filtro de bucle adaptativo se habilita y puede aplicarse a componentes de color luma (Y), croma azul (Cb), o croma rojo (Cr) en un grupo de mosaicos. El `tile_group_alf_enabled_flag` se establece igual a cero para especificar que el filtro de bucle adaptativo se inhabilita para todos los componentes de color en un grupo de mosaicos. El `tile_group_aps_id` especifica el `adaptation_parameter_set_id` del APS referido por el grupo de mosaicos. El `TemporalId` de la unidad de NAL de APS que tiene `adaptation_parameter_set_id` igual a `tile_group_aps_id` será menor o igual al `TemporalId` de la unidad de NAL de grupo de mosaicos codificada. Cuando múltiples APS con el mismo valor de `adaptation_parameter_set_id` es referido por dos o más grupos de mosaicos de la misma imagen, los múltiples APS con el mismo valor de `adaptation_parameter_set_id` pueden incluir el mismo contenido.

Los parámetros de remodelador son parámetros empleados para una herramienta de codificación de vídeo de remodelador en bucle adaptativo, que también se conoce como correlación de luma con escalado de croma (LMCS). Un ejemplo de sintaxis y semántica de remodelador de SPS son de la siguiente manera:

seq_parameter_set_rbsp() {	Descriptor
...	
sps_reshaper_enabled_flag	u(1)
...	
}	

El `sps_reshaper_enabled_flag` se establece igual a uno para especificar que el remodelador se usa en la secuencia de vídeo codificada (CVS). El `sps_reshaper_enabled_flag` se establece igual a cero para especificar que no se usa remodelador en el CVS.

Un ejemplo de sintaxis y semántica de remodelador de encabezado de grupo de mosaicos/encabezado de segmento es de la siguiente manera:

tile_group_header() {	Descriptor
...	
if(sps_reshaper_enabled_flag) {	
tile_group_reshaper_model_present_flag	u(1)
if(tile_group_reshaper_model_present_flag)	
tile_group_reshaper_model()	

tile_group_header() {	Descriptor
tile_group_reshaper_enable_flag	u(1)
if(tile_group_reshaper_enable_flag && !(qtbtt_dual_tree_intra_flag && tile_group_type == I))	
tile_group_reshaper_chroma_residual_scale_flag	u(1)
}	
...	
}	

El tile_group_reshaper_model_present_flag se establece igual uno para especificar que el tile_group_reshaper_model() está presente en el encabezado de grupo de mosaicos. El tile_group_reshaper_model_present_flag se establece igual cero para especificar que el tile_group_reshaper_model() no está presente en el encabezado de grupo de mosaicos. Cuando tile_group_reshaper_model_present_flag no está presente, se infiere que el indicador es igual a cero. El tile_group_reshaper_enabled_flag se establece igual uno para especificar que el remodelador se habilita para el grupo de mosaicos actual. El tile_group_reshaper_enabled_flag se establece igual a cero para especificar que el remodelador no se habilita para el grupo de mosaicos actual. Cuando tile_groupreshaper_enable_flag no está presente, se infiere que el indicador es igual a cero. El tile_group_reshaper_chroma_residual_scale_flag se establece igual uno para especificar que escalado residual de croma se habilita para el grupo de mosaicos actual. El tile_group_reshaper_chroma_residual_scale_flag se establece igual a cero para especificar que escalado residual de croma no se habilita para el grupo de mosaicos actual. Cuando tile_group_reshaper_chroma_residual_scale_flag no está presente, se infiere que el indicador es igual a cero.

Un ejemplo de sintaxis y semántica de modelo de remodelador de encabezado de grupo de mosaicos/encabezado de segmento es de la siguiente manera:

tile_group_reshaper_model() {	Descriptor
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW[i]	u(v)
if(reshaper_model_bin_delta_abs_CW[i] > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

El reshape_model_min_bin_idx especifica el índice de contenedor mínimo (o pedazo) a usar en el proceso de construcción de remodelador. El valor de reshape_model_min_bin_idx puede estar en el intervalo de cero y MaxBinIdx, inclusive. El valor de MaxBinIdx puede ser igual a quince. El reshape_model_delta_max_bin_idx especifica el índice de contenedor permitido máximo (o pedazo) MaxBinIdx menos el índice de contenedor máximo a usar en el proceso de construcción de remodelador. El valor de reshape_model_max_bin_idx se establece igual a MaxBinIdx menos reshape_model_delta_max_bin_idx. El reshaper_model_bin_delta_abs_cw_prec_minus1 más uno especifica el número de bits usado para la representación de la sintaxis reshape_model_bin_delta_abs_CW[i]. El reshape_model_bin_delta_abs_CW[i] especifica el valor absoluto de palabra clave delta para el contenedor i-ésimo.

El reshaper_model_bin_delta_sign_CW_flag[i] especifica el signo de reshape_model_bin_delta_abs_CW[i] de la siguiente manera. Si reshape_model_bin_delta_sign_CW_flag[i] es igual a cero, la variable correspondiente RspDeltaCW[i] es un valor positivo. De lo contrario, (p. ej., reshape_model_bin_delta_sign_CW_flag[i] no es igual a cero), la variable correspondiente RspDeltaCW[i] es un valor negativo. Cuando

reshape_model_bin_delta_sign_CW_flag[i] no está presente, se infiere que el indicador es igual a cero. La variable RspDeltaCW[i] se establece igual a $(1 - 2 * \text{reshape_model_bin_delta_sign_CW}[i]) * \text{reshape_model_bin_delta_abs_CW}[i]$.

- 5 La variable RspCW[i] se derivar como sigue. La variable OrgCW se establece igual a $(1 \ll \text{BitDepthY}) / (\text{MaxBinIdx} + 1)$. Si $\text{reshaper_model_min_bin_idx} \leq i \leq \text{reshaper_model_max_bin_idx}$ RspCW[i] = OrgCW + RspDeltaCW[i]. De otro modo, RspCW[i] = cero. El valor de RspCW[i] estará en el intervalo de treinta dos a $2 * \text{OrgCW} - 1$ si el valor de BitDepth Y es igual a diez. Las variables InputPivot[i] con i en el intervalo de 0 a MaxBinIdx + 1, inclusive se derivan de la siguiente manera. InputPivot[i] = i * OrgCW. La variable ReshapePivot[i] con i en el intervalo de 0 a MaxBinIdx + 1, inclusive, la variable ScaleCoef[i] y InvScaleCoeff[i] con i en el intervalo de cero a MaxBinIdx, inclusive, se derivan de la siguiente manera:

```

shiftY = 14
ReshapePivot[ 0 ] = 0;
for( i = 0; i <= MaxBinIdx ; i++) {
    ReshapePivot[ i + 1 ] = ReshapePivot[ i ] + RspCW[ i ]
    ScaleCoef[ i ] = ( RspCW[ i ] * (1 << shiftY) + (1 << (Log2(OrgCW) - 1))) >> (Log2(OrgCW))
    if ( RspCW[ i ] == 0 )
        InvScaleCoeff[ i ] = 0
    else
        InvScaleCoeff[ i ] = OrgCW * (1 << shiftY) / RspCW[ i ]
}

```

- 15 La variable ChromaScaleCoef[i] con i en el intervalo de 0 a MaxBinIdx, inclusive, se derivan de la siguiente manera:

```

ChromaResidualScaleLut[64] = {16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192, 8192,
8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731,
2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489, 1489,
1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024,
1024, 1024, 1024};
shiftC = 11
if ( RspCW[ i ] == 0 )
    ChromaScaleCoef[ i ] = (1 << shiftC)
Otherwise (RspCW[ i ] != 0),
    ChromaScaleCoef[ i ] = ChromaResidualScaleLut[ RspCW[ i ] >> 1 ]

```

- 20 Las propiedades de los parámetros de remodelador se pueden caracterizar de la siguiente manera. El tamaño de un conjunto de los parámetros de remodelador contenidos en la estructura de sintaxis de tile_group_reshaper_model() está usualmente alrededor de sesenta a cien bits. El modelo de remodelador es usualmente actualizado por el codificador aproximadamente una vez por segundo, que incluye muchos fotogramas. Además, los parámetros de un modelo de remodelador actualizado es improbable que sean exactamente iguales que los parámetros de un caso anterior del modelo de remodelador.

Los sistemas de codificación de vídeo anteriores incluyen ciertos problemas. Primero, tales sistemas únicamente se configuran para llevar parámetros de ALF en el APS. Además, parámetros de remodelador/LMCS se pueden compartir por múltiples imágenes y pueden incluir muchas variaciones.

En esta memoria se describen diversos mecanismos para modificar el APS para soportar eficiencia de codificación aumentada. En un primer ejemplo, se describen múltiples tipos de APS. Específicamente, un APS de tipo ALF, denominado un APS de ALF, puede contener parámetros de ALF. Además, un APS de tipo lista de escalado,

denominado APS de lista de escalado, puede contener parámetros de lista de escalado. Adicionalmente, un APS de tipo LMCS, denominado APS de LMCS, puede contener parámetros de LMCS/remodelador. El APS de ALF, APS de lista de escalado y el APS de LMCS pueden codificarse cada uno como tipos de NAL separados, y por tanto incluidos en diferentes unidades de NAL. De esta manera, un cambio a datos en un tipo de APS (p. ej., parámetros de ALF) no tiene como resultado una codificación redundante de otros tipos de datos que no cambian (p. ej., parámetros de LMCS). Por consiguiente, proporcionar múltiples tipos de APS aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador.

En un segundo ejemplo, cada APS incluye un identificador de APS (ID). Además, cada tipo de APS incluye un espacio de valores separado para ID de APS correspondientes. Tales espacios de valores pueden superponerse. Por consiguiente, un APS de un primer tipo (p. ej. un APS de ALF) puede incluir el mismo ID de APS como APS de un segundo tipo (p. ej. un APS de LMCS). Esto se consigue al identificar cada APS por una combinación de un tipo de parámetro de APS y un ID de APS. Al permitir que cada tipo de APS incluya un espacio de valores diferente, el códec no necesita comprobar conflictos de ID en tipos de APS. Además, al permitir que los espacios de valor se superpongan, el códec puede evitar emplear valores de ID más grandes, lo que tiene como resultado ahorros de bits. Como tal, emplear espacios de valor superpuestos separados para los APS de diferentes tipos aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador.

En un tercer ejemplo, parámetros de LMCS se incluyen en un APS de LMCS. Como se señala anteriormente, parámetros de LMCS/remodelador pueden cambiar aproximadamente una vez por segundo. Las secuencias de vídeo pueden visualizar de treinta a sesenta imágenes por segundo. Como tal, parámetros de LMCS pueden no cambiar para treinta a sesenta fotogramas. Incluir parámetros de LMCS en un APS de LMCS reduce significativamente la codificación redundante de parámetros de LMCS. Un encabezado de segmento y/o un encabezado de imagen asociados con un segmento pueden referirse al APS de LMCS relevante. De esta manera, los parámetros de LMCS únicamente se codifican cuando cambian los parámetros de LMCS un segmento. Por consiguiente, emplear un APS de LMCS para codificar parámetros LMCS aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador.

La FIGURA 1 es un diagrama de flujo de un método operativo de ejemplo 100 para codificar una señal de vídeo. Específicamente, una señal de vídeo se codifica en un codificador. El proceso de codificación comprime la señal de vídeo empleando diversos mecanismos para reducir el tamaño del archivo de vídeo. Un tamaño de archivo más pequeño permite que el archivo de vídeo comprimido se transmita hacia un usuario, al tiempo que reduce la sobrecarga de ancho de banda asociada. El decodificador luego decodifica el archivo de vídeo comprimido para reconstruir la señal de vídeo original para mostrarla a un usuario final. El proceso de decodificación generalmente refleja el proceso de codificación para permitir que el decodificador reconstruya consistentemente la señal de vídeo.

En la etapa 101, la señal de vídeo se introduce en el codificador. Por ejemplo, la señal de vídeo puede ser un archivo de vídeo sin comprimir almacenado en la memoria. Como otro ejemplo, el archivo de vídeo puede ser capturado por un dispositivo de captura de vídeo, como una videocámara, y codificado para soportar la retransmisión 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 fotogramas de imagen que, cuando se ven en una secuencia, dan la impresión visual de movimiento. Los fotogramas contienen píxeles que se expresan en términos de luz, denominados en esta memoria componentes de luma (o muestras de luma), y color, que se denomina componentes de croma (o muestras de color). En algunos ejemplos, los fotogramas también pueden contener valores de profundidad para admitir la visualización tridimensional.

En la etapa 103, el vídeo se particiona en bloques. El particionamiento incluye subdividir los píxeles de cada fotograma en bloques cuadrados y/o rectangulares para la compresión. Por ejemplo, en codificación de vídeo de alta eficiencia (HEVC) (también conocida como H.265 y MPEG-H Parte 2) el fotograma se puede dividir primero en unidades de árbol de codificación (CTU), que son bloques de un tamaño predefinido (por ejemplo, sesenta y cuatro píxeles por sesenta y cuatro píxeles). Las CTU contienen ambas muestras de luma y croma. Pueden emplearse árboles de codificación para dividir las CTU en bloques y luego subdividir recursivamente los bloques hasta que se logren configuraciones que admitan una codificación adicional. Por ejemplo, los componentes de luma de un fotograma pueden subdividirse hasta que los bloques individuales contengan valores de iluminación relativamente homogéneos. Además, los componentes de croma de un fotograma pueden subdividirse hasta que los bloques individuales contengan valores de color relativamente homogéneos. Por consiguiente, los mecanismos de partición varían según el contenido de los fotogramas de vídeo.

En la etapa 105, se emplean diversos mecanismos de compresión para comprimir los bloques de imagen particionados en la etapa 103. Por ejemplo, se puede emplear interpolación y/o intrapredicción. La interpolación se diseña para aprovechar el hecho de que los objetos en una escena común tienden a aparecer en fotogramas sucesivos. Por consiguiente, un bloque que representa un objeto en un fotograma de referencia no necesita describirse repetidamente en fotogramas adyacentes. Específicamente, un objeto, como una mesa, puede

permanecer en una posición constante en varios fotogramas. Por lo tanto, la tabla se describe una vez y los fotogramas adyacentes pueden hacer referencia al fotograma de referencia. Pueden emplearse mecanismos de coincidencia de patrones para hacer coincidir objetos en múltiples fotogramas. Además, los objetos en movimiento se pueden representar en varios fotogramas, por ejemplo, debido al movimiento del objeto o al movimiento de la cámara. Como ejemplo particular, un vídeo puede mostrar un automóvil que se mueve por la pantalla en múltiples fotogramas. Se pueden emplear vectores de movimiento para describir dicho movimiento. Un vector de movimiento es un vector bidimensional que proporciona un desplazamiento de las coordenadas de un objeto en un fotograma a las coordenadas del objeto en un fotograma de referencia. Como tal, la interpredicción puede codificar un bloque de imagen en un fotograma actual como un conjunto de vectores de movimiento que indican un desplazamiento de un bloque correspondiente en un fotograma de referencia.

La intrapredicción codifica bloques en un fotograma común. La intrapredicción aprovecha el hecho de que los componentes de luma y croma tienden a agruparse en un fotograma. Por ejemplo, un parche de verde en una parte de un árbol tiende a colocarse adyacente a parches de verde similares. La intrapredicción emplea múltiples modos de predicción direccional (por ejemplo, treinta y tres en HEVC), un modo plano y un modo actual directo (DC). Los modos direccionales indican que un bloque actual es similar/igual que las muestras de un bloque vecino en la dirección correspondiente. El modo plano indica que una serie de bloques a lo largo de una fila/columna (por ejemplo, un plano) se puede interpolar en función de los 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 al emplear una pendiente relativamente constante en los valores cambiantes. El modo DC se emplea para suavizado de fronteras e indica que un bloque es similar/igual que un valor promedio asociado con muestras de todos los bloques vecinos asociados con las direcciones angulares de los modos de predicción direccional. Por consiguiente, los bloques de intrapredicción pueden representar bloques de imagen como diversos valores de modo de predicción relacional en lugar de los valores reales. Además, los bloques de interpredicción pueden representar bloques de imagen como valores de vector de movimiento en lugar de valores reales. En cualquier caso, es posible que los bloques de predicción no representen exactamente los bloques de imágenes en algunos casos. Cualesquiera diferencias se almacenan en bloques residuales. Se pueden aplicar transformaciones a los bloques residuales para comprimir aún más el archivo.

En la etapa 107, se pueden aplicar diversas técnicas de filtrado. En HEVC, los filtros se aplican según un esquema de filtrado en bucle. La predicción basada en bloques discutida anteriormente puede resultar en la creación de imágenes en bloques en el decodificador. Además, el esquema de predicción basado en bloques puede codificar un bloque y luego reconstruir el bloque codificado para su uso posterior como bloque de referencia. El esquema de filtrado en bucle aplica de forma iterativa filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptativo y filtros de compensación adaptativa de muestra (SAO) a los bloques/ fotogramas. Estos filtros mitigan tales artefactos de bloqueo para que el archivo codificado se pueda reconstruir con precisión. Además, estos filtros mitigan los artefactos en los bloques de referencia reconstruidos, de modo que es menos probable que los artefactos creen artefactos adicionales en los bloques posteriores que se codifican en función de los bloques de referencia reconstruidos.

Una vez que la señal de vídeo ha sido particionada, comprimida y filtrada, los datos resultantes se codifican en un flujo de bits en la etapa 109. El flujo de bits incluye los datos discutidos anteriormente, así como cualquier dato de señalización deseado para soportar la reconstrucción adecuada de la señal de vídeo en el decodificador. Por ejemplo, tales datos pueden incluir datos de partición, datos de predicción, bloques residuales y diversos indicadores que proporcionan instrucciones de codificación al decodificador. El flujo de bits puede almacenarse en la memoria para su transmisión hacia un decodificador cuando se solicite. El flujo de bits también se puede difundir y/o multidifundir hacia una pluralidad de decodificadores. La creación del flujo de bits es un proceso iterativo. Por consiguiente, las etapas 101, 103, 105, 107 y 109 pueden ocurrir de forma continua y/o simultánea en muchos fotogramas y bloques. El orden mostrado en la FIGURA 1 se presenta para mayor claridad y facilidad de discusión, y no pretende limitar el proceso de codificación de vídeo a un orden en particular.

El decodificador recibe el flujo de bits y comienza el proceso de decodificación en la etapa 111. Específicamente, el decodificador emplea un esquema de decodificación por entropía para convertir el flujo de bits en la sintaxis y los datos de vídeo correspondientes. El decodificador emplea los datos de sintaxis del flujo de bits para determinar las particiones para los fotogramas en la etapa 111. La partición debe coincidir con los resultados de la partición de bloque en la etapa 103. Ahora se describe la codificación/decodificación por entropía empleada en la etapa 111. El codificador hace muchas elecciones durante el proceso de compresión, como seleccionar esquemas de partición de bloque entre varias opciones posibles basadas en el posicionamiento espacial de los valores en las imágenes de entrada. La señalización de las opciones exactas puede emplear una gran cantidad de contenedores. Como se usa en esta memoria, un contenedor es un valor binario que se trata como una variable (por ejemplo, un valor de bit que puede variar según el contexto). La codificación por entropía permite al codificador descartar cualquier opción que claramente no sea viable para un caso particular, dejando un conjunto de opciones permitidas. A cada opción permitida se le asigna una palabra de código. La longitud de las palabras de código se basa en el número de opciones permitidas (por ejemplo, un contenedor para dos opciones, dos contenedores para tres o cuatro opciones, etc.). El codificador luego codifica la palabra de código para la opción seleccionada. Este esquema reduce el tamaño de las palabras de código ya que las palabras de código son tan grandes como se desee para

indicar de forma única una selección de un pequeño subconjunto de opciones permitidas en lugar de indicar de forma única la selección de un conjunto potencialmente grande de todas las opciones posibles. El decodificador luego decodifica la selección determinando el conjunto de opciones permitidas de manera similar al codificador. Al determinar el conjunto de opciones permitidas, el decodificador puede leer la palabra de código y determinar la selección realizada por el codificador.

En la etapa 113, el decodificador realiza la decodificación de bloques. Específicamente, el decodificador emplea transformadas inversas para generar bloques residuales. Luego, el decodificador emplea los bloques residuales y los bloques de predicción correspondientes para reconstruir los bloques de imagen según la partición. Los bloques de predicción pueden incluir tanto bloques de intrapredicción como bloques de interpredicción como se generan en el codificador en la etapa 105. A continuación, los bloques de imágenes reconstruidos se colocan en fotogramas de una señal de vídeo reconstruida según los datos de partición determinados en la etapa 111. La sintaxis para la etapa 113 también se puede señalar en el flujo de bits mediante la codificación por entropía como se ha discutido anteriormente.

En la etapa 115, se realiza el filtrado en los fotogramas de la señal de vídeo reconstruida de una manera similar a la etapa 107 en el codificador. Por ejemplo, se pueden aplicar filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptativo y filtros SAO a los fotogramas para eliminar los artefactos de bloque. Una vez que se filtran los fotogramas, la señal de vídeo se puede sacar a una pantalla en la etapa 117 para que la vea un usuario final.

La FIGURA 2 es un diagrama esquemático de un ejemplo de sistema de codificación y decodificación (códec) 200 para codificación de vídeo. Específicamente, el sistema de códec 200 proporciona funcionalidad para respaldar la implementación del método operativo 100. El sistema de códec 200 se generaliza para representar los componentes empleados tanto en un codificador como en un decodificador. El sistema de códec 200 recibe y particiona una señal de vídeo como se ha discutido con respecto a las etapas 101 y 103 en el método operativo 100, lo que da como resultado una señal de vídeo particionada 201. A continuación, el sistema de códec 200 comprime la señal de vídeo particionada 201 en un flujo de bits codificado cuando actúa como codificador como se describe con respecto a las etapas 105, 107 y 109 en el método 100. Cuando actúa como decodificador, el sistema de códec 200 genera una señal de vídeo de salida a partir del flujo de bits como se describe con respecto a las etapas 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 cuantización y de escalado de transformada 213, un componente de estimación intraimagen 215, un componente de predicción intraimagen 217, un componente de compensación de movimiento 219, un componente de estimación de movimiento 221, un componente de transformada inversa y de escalado 229, un componente de análisis de control de filtro 227, un componente de filtros en bucle 225, un componente de búfer de imágenes decodificadas 223 y un componente de codificación aritmética binaria adaptativa de contexto (CABAC) y de formateo de encabezado 231. Dichos componentes se acoplan como se muestra. En la FIGURA 2, las líneas negras indican el movimiento de los datos a codificar/decodificar, mientras que las líneas discontinuas indican el movimiento de los datos de control que controlan el funcionamiento de otros componentes. Todos 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 intraimagen 217, el componente de compensación de movimiento 219, el componente de transformada inversa y de escalado 229, el componente de filtros en bucle 225 y el componente de búfer de imágenes decodificadas 223. Estos componentes se describen ahora.

La señal de vídeo particionada 201 es una secuencia de vídeo capturada que ha sido particionada en bloques de píxeles por un árbol de codificación. Un árbol de codificación emplea diversos modos de partir para subdividir un bloque de píxeles en bloques de píxeles más pequeños. Estos bloques pueden luego subdividirse en bloques más pequeños. Los bloques pueden denominarse nodos en el árbol de codificación. Los nodos principales más grandes se parten en nodos secundarios más pequeños. El número de veces que se subdivide un nodo se denomina profundidad del árbol de codificación/nodo. Los bloques divididos se pueden incluir en unidades de codificación (CU) en algunos casos. Por ejemplo, una CU puede ser una subparte de una CTU que contiene un bloque de luma, un bloque(s) de croma (Cr) de diferencia de rojo y bloque(s) de croma de diferencia de azul (Cb) junto con las instrucciones de sintaxis correspondientes para la CU. Los modos de partir pueden incluir un árbol binario (BT), árbol triple (TT) y un árbol cuádruple (QT) empleados para particionar un nodo en dos, tres o cuatro nodos secundarios, respectivamente, de diferentes formas dependiendo de los modos de partir empleados. La señal de vídeo particionada 201 se reenvía al componente de control de codificador general 211, al componente de cuantización y de escalado de transformada 213, al componente de estimación intraimagen 215, al componente de análisis de control de filtro 227 y al componente de estimación de movimiento 221 para su compresión.

El componente de control de codificador general 211 se configura para tomar decisiones relacionadas con la codificación de las imágenes de la secuencia de vídeo en el flujo de bits según restricciones de aplicación. Por ejemplo, el componente de control de codificador general 211 gestiona la optimización del tamaño de tasa de bits / flujo de bits frente a la calidad de reconstrucción. Dichas decisiones se pueden tomar en función de la disponibilidad de espacio de almacenamiento/ancho de banda y las solicitudes de resolución de imagen. El

componente de control de codificador general 211 también gestiona la utilización de búfer a la luz de la velocidad de transmisión para mitigar los problemas de saturación y falta de ejecución del búfer. Para gestionar estos problemas, el componente de control de codificador general 211 gestiona la partición, la predicción y el filtrado de los otros componentes. Por ejemplo, el componente de control de codificador general 211 puede aumentar dinámicamente la complejidad de la compresión para aumentar la resolución y aumentar el uso del ancho de banda o disminuir la complejidad de la compresión para disminuir la resolución y el uso del ancho de banda. Por 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 la señal de vídeo con las preocupaciones sobre la tasa de bits. El componente de control de codificador general 211 crea datos de control, que controlan el funcionamiento de los otros componentes. Los datos de control también se reenvían al componente de CABAC y de formateo de encabezado 231 para codificarse en el flujo de bits para señalar parámetros para decodificar en el decodificador.

La señal de vídeo particionada 201 también se envía al componente de estimación de movimiento 221 y al componente de compensación de movimiento 219 para la interpredicción. Un fotograma o segmento de la señal de vídeo particionada 201 puede dividirse en múltiples bloques de vídeo. El componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 realizan una codificación interpredictiva del bloque de vídeo recibido en relación con uno o más bloques en uno o más fotogramas de referencia para proporcionar predicción temporal. El sistema de códec 200 puede realizar múltiples pasadas 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 sumamente integrados, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por el componente de estimación de movimiento 221, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un objeto codificado con respecto a un bloque predictivo. Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque a codificar, en términos de diferencia de píxeles. Un bloque predictivo también puede denominarse bloque de referencia. Dicha diferencia de píxeles puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. HEVC emplea varios objetos codificados que incluyen una CTU, bloques de árbol de codificación (CTB) y CU. Por ejemplo, una CTU se puede dividir en varios CTB, que luego se pueden dividir en CB para su inclusión en las CU. Una CU puede codificarse 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, PU y TU 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/fotograma actual, y puede seleccionar los bloques de referencia, vectores de movimiento, etc. que tienen las mejores características de distorsión de tasa. Las mejores características de distorsión de tasa equilibran tanto la calidad de la reconstrucción de vídeo (por ejemplo, la cantidad de datos perdidos por compresión) con la eficiencia de codificación (por ejemplo, el tamaño de la codificación final).

En algunos ejemplos, el sistema de códec 200 puede calcular valores para posiciones de píxeles subentero de imágenes de referencia almacenadas en el componente de búfer de imágenes decodificadas 223. Por ejemplo, el sistema de códec de vídeo 200 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionarios de la imagen de referencia. Por lo tanto, el componente de estimación de movimiento 221 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completos y las posiciones de píxeles fraccionarios y sacar un vector de movimiento con precisión de píxeles fraccionarios. El componente de estimación de movimiento 221 calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento intercodificado 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 tiene como salida el vector de movimiento calculado como datos de movimiento al componente de CABAC y de formateo de encabezado 231 para codificación y movimiento al componente de compensación de movimiento 219.

La compensación de movimiento, realizada por el componente de compensación de movimiento 219, puede implicar buscar o generar el bloque predictivo en función del vector de movimiento determinado por el componente de estimación de movimiento 221. De nuevo, el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 pueden integrarse funcionalmente, en algunos ejemplos. Al 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 que apunta el vector de movimiento. A continuación, se forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles. En general, el componente de estimación de movimiento 221 realiza una estimación de movimiento con relación a los componentes de luma, y el componente de compensación de movimiento 219 usa vectores de movimiento calculados en función de los componentes de luma tanto para los componentes de croma como para los componentes de luma. El bloque predictivo y el bloque residual se reenvían al componente de cuantización y de escalado de transformada 213.

La señal de vídeo particionada 201 también se envía al componente de estimación intraimagen 215 y al componente de predicción intraimagen 217. Al igual que con el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219, el componente de estimación intraimagen 215 y el componente de predicción intraimagen 217 pueden estar sumamente integrados, pero se ilustran por separado con fines conceptuales. El componente de estimación intraimagen 215 y el componente de predicción intraimagen 217 intrapredicen un bloque actual en relación con los bloques en un fotograma actual, como alternativa a la interpredicción realizada por el componente de estimación de movimiento 221 y el componente de compensación de movimiento 219 entre fotogramas, como se ha descrito anteriormente. En particular, el componente de estimación intraimagen 215 determina un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, el componente de estimación intraimagen 215 selecciona un modo de intrapredicción apropiado para codificar un bloque actual a partir de múltiples modos de intrapredicción probados. Los modos de intrapredicción seleccionados se reenvían luego al componente de CABAC y de formateo de encabezado 231 para su codificación.

Por ejemplo, el componente de estimación intraimagen 215 calcula los valores de distorsión de tasa usando un análisis de distorsión de tasa para los diversos modos de intrapredicción probados y selecciona el modo de intrapredicción que tiene 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, una cantidad de bits) utilizada para producir el bloque codificado. El componente de estimación intraimagen 215 calcula las relaciones a partir de las distorsiones y tasas para los diversos bloques codificados para determinar qué modo de intrapredicción presenta el mejor valor de distorsión de tasa para el bloque. Además, el componente de estimación intraimagen 215 puede configurarse para codificar bloques de profundidad de un mapa de profundidad utilizando un modo de modelado de profundidad (DMM) basado en la optimización de la distorsión de tasa (RDO).

El componente de predicción intraimagen 217 puede generar un bloque residual a partir del bloque predictivo basado en los modos de intrapredicción seleccionados determinados por el componente de estimación intraimagen 215 cuando se implementa en un codificador o leer el bloque residual del flujo de bits cuando se implementa en un decodificador. El bloque residual incluye la diferencia de valores entre el bloque predictivo y el bloque original, representado como una matriz. A continuación, el bloque residual se reenvía al componente de cuantización y de escalado de transformada 213. El componente de estimación intraimagen 215 y el componente de predicción intraimagen 217 pueden operar tanto en componentes de luma como de croma.

El componente de cuantización y de escalado de transformada 213 se configura para comprimir más el bloque residual. El componente de cuantización y de escalado de transformada 213 aplica una transformada, tal como una transformada de coseno discreta (DCT), una transformada de seno discreta (DST) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficiente de transformada residuales. También se podrían utilizar transformadas de ondícula, transformadas de enteros, transformada de subbandas u otros tipos de transformadas. La transformada puede convertir la información residual de un dominio de valor de píxel en un dominio de transformada, tal como un dominio de frecuencia. El componente de cuantización y de escalado de transformada 213 también se configura para escalador la información residual transformada, por ejemplo, basándose en la frecuencia. Tal escalado implica aplicar un factor de escalado a la información residual para que la información de frecuencia diferente se cuantice en diferentes granularidades, lo que puede afectar la calidad visual final del vídeo reconstruido. El componente de cuantización y de escalado de transformada 213 también se configura para cuantizar los coeficientes de transformada para reducir aún más la tasa de bits. El proceso de cuantización puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantización se puede modificar ajustando un parámetro de cuantización. En algunos ejemplos, el componente de cuantización y de escalado de transformada 213 puede entonces realizar una exploración de la matriz que incluye los coeficientes de transformada cuantizados. Los coeficientes de transformada cuantizados se reenvían al componente de CABAC y de formateo de encabezado 231 para codificarse en el flujo de bits.

El componente de transformada inversa y de escalado 229 aplica una operación inversa del componente de cuantización y de escalado de transformada 213 para soportar la estimación del movimiento. El componente de transformada inversa y de escalado 229 aplica escalado inverso, transformación y/o cuantización para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para uso posterior como un bloque de referencia que puede convertirse 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 añadiendo el bloque residual de nuevo a un bloque predictivo correspondiente para su uso en la estimación de movimiento de un bloque/fotograma posterior. Filtros se aplican a los bloques de referencia reconstruidos para mitigar los artefactos creados durante el escalado, la cuantización y la transformada. De lo contrario, tales artefactos podrían causar una predicción inexacta (y crear artefactos adicionales) cuando se predicen bloques posteriores.

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 los bloques de imagen reconstruidos. Por ejemplo, el bloque residual transformado del

componente de transformada inversa y de escalado 229 puede combinarse con un bloque de predicción correspondiente del componente de predicción intraimagen 217 y/o el componente de compensación de movimiento 219 para reconstruir el bloque de imagen original. A continuación, los filtros se pueden aplicar al bloque de imagen reconstruido. En algunos ejemplos, los filtros se pueden aplicar en cambio a los bloques residuales. Como ocurre con otros componentes de la FIGURA2, el componente de análisis de control de filtro 227 y el componente de filtros en bucle 225 están sumamente integrados y pueden implementarse juntos, pero se representan por separado con fines conceptuales. Los filtros aplicados a los bloques de referencia reconstruidos se aplican 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 dónde deberían aplicarse dichos filtros y establece los parámetros correspondientes. Dichos datos se reenvían al componente de CABAC y de formateo de encabezado 231 como datos de control de filtro para la codificación. El componente de filtros en bucle 225 aplica dichos filtros basándose 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 se pueden aplicar en el dominio espacial / de píxeles (por ejemplo, en un bloque de píxeles reconstruido) o en el dominio de frecuencia, según el ejemplo.

Cuando funciona como codificador, el bloque de imagen reconstruido filtrado, el bloque residual y/o el bloque de predicción se almacenan en el componente de búfer de imágenes decodificadas 223 para su uso posterior en la estimación de movimiento como se ha discutido anteriormente. Cuando funciona como decodificador, el componente de búfer de imágenes decodificadas 223 almacena y reenvía los bloques reconstruidos y filtrados hacia una pantalla como parte de una señal de video de salida. El componente de búfer de imágenes decodificadas 223 puede ser cualquier dispositivo de memoria capaz de almacenar bloques de predicción, bloques residuales y/o bloques de imágenes reconstruidas.

El componente de CABAC y de formateo de encabezado 231 recibe los datos de los diversos componentes del sistema de códec 200 y codifica dichos datos en un flujo de bits codificado para su transmisión hacia un decodificador. Específicamente, el componente de CABAC y de formateo de encabezado 231 genera diversos encabezados para codificar datos de control, tales como datos de control general y datos de control de filtro. Además, los datos de predicción, incluidos los datos de intrapredicción y de movimiento, así como los datos residuales en forma de datos de coeficientes de transformada cuantizados, están todos codificados en el flujo de bits. El flujo de bits final incluye toda la información deseada por el decodificador para reconstruir la señal de video particionada original 201. Dicha información también puede incluir tablas de índice de modo de intrapredicción (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques, indicaciones de los modos de intrapredicción más probables, una indicación de información de partición, etc. Tales datos pueden codificarse empleando codificación por entropía. Por ejemplo, la información puede codificarse empleando codificación de longitud variable adaptativa al contexto (CAVLC), CABAC, codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación por entropía de partición de intervalo de probabilidad (PIPE) u otra técnica de codificación por entropía. Después de la codificación por entropía, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, un decodificador de video) o archivar para su posterior transmisión o recuperación.

La FIGURA 3 es un diagrama de bloques que ilustra un codificador de video de ejemplo 300. Puede emplearse el codificador de video 300 para implementar las funciones de codificación del sistema de códec 200 y/o implementar las etapas 101, 103, 105, 107 y/o 109 del método operativo 100. El codificador 300 particiona una señal de video de entrada, dando como resultado una señal de video particionada 301, que es sustancialmente similar a la señal de video particionada 201. La señal de video particionada 301 es luego comprimida y codificada en un flujo de bits por componentes del codificador 300.

Específicamente, la señal de video particionada 301 se reenvía a un componente de predicción intraimagen 317 para la intrapredicción. El componente de predicción intraimagen 317 puede ser sustancialmente similar al componente de estimación intraimagen 215 y al componente de predicción intraimagen 217. La señal de video particionada 301 también se reenvía a un componente de compensación de movimiento 321 para la interpredicción basada en bloques de referencia en un componente de búfer de imágenes decodificadas 323. El componente de compensación de movimiento 321 puede ser sustancialmente similar al componente de estimación de movimiento 221 y al componente de compensación de movimiento 219. Los bloques de predicción y los bloques residuales del componente de predicción intraimagen 317 y el componente de compensación de movimiento 321 se reenvían a un componente de cuantización y transformada 313 para la transformada y cuantización de los bloques residuales. El componente de cuantización y transformada 313 puede ser sustancialmente similar al componente de escalado de transformada y cuantización 213. Los bloques residuales transformados y cuantizados y los bloques de predicción correspondientes (junto con los datos de control asociados) se reenvían a un componente de codificación por entropía 331 para codificar en un flujo de bits. El componente de codificación por entropía 331 puede ser sustancialmente similar al componente de CABAC y de formateo de encabezado 231.

Los bloques residuales transformados y cuantizados y/o los correspondientes bloques de predicción también se reenvían desde el componente de cuantización y transformada 313 a un componente de cuantización y transformada inversa 329 para la reconstrucción en bloques de referencia para su uso por el componente de

compensación de movimiento 321. El componente de cuantización y transformada inversa 329 puede ser sustancialmente similar al componente de transformada inversa y de escalado 229. Los filtros en bucle en un componente de filtros en bucle 325 también se aplican a los bloques residuales y/o bloques de referencia reconstruidos, según el ejemplo. El componente de filtros en bucle 325 puede ser sustancialmente similar al componente de análisis de control de filtro 227 y al componente de filtros en bucle 225. El componente de filtros en bucle 325 puede incluir múltiples filtros como se describe con respecto al componente de filtros en bucle 225. A continuación, los bloques filtrados se almacenan en un componente de búfer de imágenes decodificadas 323 para su uso como bloques de referencia por el componente de compensación de movimiento 321. El componente de búfer de imágenes decodificadas 323 puede ser sustancialmente similar al componente de búfer de imágenes decodificadas 223.

La FIGURA 4 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo 400. El decodificador de vídeo 400 se puede emplear para implementar las funciones de decodificación del sistema de códec 200 y/o implementar las etapas 111, 113, 115 y/o 117 del método operativo 100. El decodificador 400 recibe un flujo de bits, por ejemplo, de un codificador 300, y genera una señal de vídeo de salida reconstruida basada en el flujo de bits para su visualización para un usuario final.

El flujo de bits es recibido por un componente de decodificación por entropía 433. El componente de decodificación por entropía 433 se configura para implementar un esquema de decodificación por entropía, tal como CAVLC, CABAC, SBAC, codificación PIPE u otras técnicas de codificación por entropía. Por ejemplo, el componente de decodificación por entropía 433 puede emplear información de encabezado para proporcionar un contexto para interpretar datos adicionales codificados como palabras de código en el flujo de bits. La información decodificada incluye cualquier información deseada para decodificar la señal de vídeo, tal como datos de control general, datos de control de filtro, información de partición, datos de movimiento, datos de predicción y coeficientes de transformada cuantizados de bloques residuales. Los coeficientes de transformada cuantizados se reenvían a un componente de cuantización y transformada inversa 429 para su reconstrucción en bloques residuales. El componente de cuantización y transformada inversa 429 puede ser similar al componente de cuantización y transformada inversa 329.

Los bloques residuales reconstruidos y/o los bloques de predicción se reenvían al componente de predicción intraimagen 417 para su reconstrucción en bloques de imagen basados en operaciones de intrapredicción. El componente de predicción intraimagen 417 puede ser similar al componente de estimación intraimagen 215 y al componente de predicción intraimagen 217. Específicamente, el componente de predicción intraimagen 417 emplea modos de predicción para localizar un bloque de referencia en el fotograma y aplica un bloque residual al resultado para reconstruir bloques de imágenes intrapredichos. Los bloques de imagen intrapredichos reconstruidos y/o los bloques residuales y los datos de interpredicción correspondientes se reenvían a un componente de búfer de imágenes decodificadas 423 a través de un componente de filtros en bucle 425, que puede ser sustancialmente similar al componente de búfer de imágenes decodificadas 223 y al componente de filtro de bucle 225, respectivamente. El componente de filtros en bucle 425 filtra los bloques de imágenes reconstruidas, los bloques residuales y/o los bloques de predicción, y dicha información se almacena en el componente de búfer de imágenes decodificadas 423. Los bloques de imágenes reconstruidas del componente de búfer de imágenes decodificadas 423 se reenvían a un componente de compensación de movimiento 421 para la interpredicción. 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 enviarse a través del componente de filtros en bucle 425 al componente de búfer de imágenes decodificadas 423. El componente de búfer de imágenes decodificadas 423 continúa almacenando bloques de imágenes reconstruidas adicionales, que pueden reconstruirse en fotogramas a través de la información de partición. Estos fotogramas también se pueden colocar en una secuencia. La secuencia se envía a una pantalla como una señal de vídeo de salida reconstruida.

La FIGURA 5 es un diagrama esquemático que ilustra un flujo de bits de ejemplo 500 que contiene múltiples tipos de APS que incluyen tipos diferentes de parámetros de herramienta de codificación. Por ejemplo, el flujo de bits 500 puede ser generado por un sistema de códec 200 y/o un codificador 300 para decodificar por un sistema de códec 200 y/o un decodificador 400. Como otro ejemplo, el flujo de bits 500 puede ser generado por un codificador en la etapa 109 del método 100 para ser utilizado por un decodificador en la etapa 111.

El flujo de bits 500 incluye un conjunto de parámetros de secuencia (SPS) 510, una pluralidad de conjuntos de parámetros de imagen (PPS) 511, una pluralidad de APS de ALF 512, una pluralidad de APS de lista de escalado 513, una pluralidad de LMC APS 514, una pluralidad de encabezados de segmento 515 y datos de imagen 520. Un SPS 510 contiene datos de secuencia comunes a todas las imágenes de la secuencia de vídeo contenida en el flujo de bits 500. Tales datos pueden incluir el tamaño de imagen, la profundidad de bits, los parámetros de herramienta de codificación, las restricciones de tasa de bits, etc. El PPS 511 contiene parámetros que se aplican a toda la imagen. Por tanto, cada imagen de una secuencia de vídeo puede referirse a un PPS 511. Cabe señalar que, si bien cada imagen se refiere a un PPS 511, un único PPS 511 puede contener datos para múltiples imágenes

en algunos ejemplos. Por ejemplo, múltiples imágenes similares pueden codificarse según parámetros similares. En tal caso, un único PPS 511 puede contener datos para tales imágenes similares. El PPS 511 puede indicar herramientas de codificación disponibles para segmentos en imágenes correspondientes, parámetros de cuantización, offsets, etc. El encabezado de segmento 515 contiene parámetros que son específicos para cada segmento en una imagen. Por tanto, puede haber un encabezado de segmento 515 por segmento en la secuencia de vídeo. El encabezado de segmento 515 puede contener información de tipo de segmento, recuentos de orden de imagen (POC), listas de imágenes de referencia, pesos de predicción, puntos de entrada de mosaico, parámetros de desbloqueo, etc. Cabe señalar que un encabezado de segmento 515 también se pueden denominar encabezado de grupo de mosaicos en algunos contextos.

Un APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a una o más imágenes 521 y/o segmentos 523. En el ejemplo mostrado, un APS puede separarse en múltiples tipos. Un APS de ALF 512 es un APS de tipo ALF que incluye parámetros de ALF. Un ALF es un filtro basado en bloque adaptativo que incluye una función de transferencia controlada por parámetros variables y emplea retroinformación de un bucle de retroinformación para refinar la función de transferencia. Además, la ALF se emplea para corregir artefactos de codificación (p. ej., errores) que ocurren como resultado de la codificación basada en bloques. Un filtro adaptativo es un filtro lineal con un controlador de función de transferencia por parámetros variables que puede ser controlado por un algoritmo de optimización, tal como el proceso de RDO que opera en el codificador. Como tal, parámetros de ALF incluidos en un APS de ALF 512 pueden incluir parámetros variables seleccionados por el codificador para provocar que el filtro retire artefactos de codificación basados en bloques durante la decodificación en el decodificador.

Un APS de lista de escalado 513 es un APS de tipo lista de escalado que incluye parámetros de lista de escalado. Como se ha discutido anteriormente, un bloque actual se codifica según interpredicción o intrapredicción que tiene como resultado un residual. El residual es la diferencia entre los valores de luma y/o de croma del bloque y valores correspondientes del bloque de predicción. Una transformada se aplica entonces al residual para convertir el residual en coeficientes de transformada (que son más pequeñas que los valores residuales). Codificar contenido de alta definición y/o ultra alta definición puede tener como resultado aumento de datos residuales. Un proceso de transformada simple puede tener como resultado ruido de cuantización significativo cuando se aplica a tales datos. Como tal, parámetros de lista de escalado incluidos en un APS de lista de escalado 513 pueden incluir parámetros de ponderación que se pueden aplicar para escalar las matrices de transformada para tener en cuenta variaciones en resoluciones de visualización y/o niveles aceptables de ruido de cuantización en una imagen de vídeo decodificado resultante.

Un APS de LMCS 514 es un APS de tipo LMCS que incluye parámetros de LMCS, que también se conoce que parámetros de remodelador. El sistema visual humano es menos capaz de distinguir diferencias en color (p. ej., cromaticidad) que diferencias en luz (p. ej., luminancia). Como tal, algunos sistemas de vídeo usan un mecanismo de submuestreo de croma para comprimir datos de vídeo al reducir la resolución de valores de croma sin ajustar los correspondientes valores de luma. Una preocupación con tales mecanismos es que la interpolación asociada puede producir valores de croma interpolados durante la decodificación que son incompatibles con correspondientes valores de luma en algunas ubicaciones. Esto crea artefactos de color en tales ubicaciones, que deben ser corregidos mediante filtros correspondientes. Esto es complicado por mecanismos de correlación luma. La correlación de luma es el proceso de volver a correlacionar componentes de luma codificados en un rango dinámico de una señal luma de entrada (p. ej., según una función lineal definida a trozos). Esto comprime los componentes de luma. Los algoritmos LMCS escalan los valores de croma comprimidos en función de las correlaciones de luma para retirar artefactos relacionados con submuestreo de croma. Como tal, parámetros de LMCS incluidos en un APS de LMCS 514 indican el escalado de croma usado para tener en cuenta correlación de luma. Los parámetros de LMCS se determinan por el codificador y se pueden emplear por el decodificador para filtrar artefactos provocados por submuestreo de croma en cuando se emplea correlación de luma.

Los datos de imagen 520 contienen datos de vídeo codificados según la interpredicción y/o intrapredicción, así como los correspondientes datos residuales transformados y cuantizados. Por ejemplo, una secuencia de vídeo incluye una pluralidad de imágenes 521 codificadas que datos de imagen. Una imagen 521 es un único fotograma de una secuencia de vídeo y por tanto generalmente se visualiza como única unidad cuando se visualiza la secuencia de vídeo. Sin embargo, pueden visualizarse imágenes parciales para implementar ciertas tecnologías tales como realidad virtual, imagen en imagen, etc. Las imágenes 521 hacen referencia a un PPS 511. Las imágenes 521 se dividen en segmentos 523. Un segmento 523 se puede definir como sección horizontal de una imagen 521. Por ejemplo, un segmento 523 puede contener una parte de la altura de la imagen 521 y la anchura completa de la imagen 521. En otros casos, la imagen 521 puede dividirse en columnas y filas y un segmento 523 se puede incluir en una parte rectangular de la imagen 521 creada por tales columnas y filas. En algunos sistemas los segmentos 523 se subdividen en mosaicos. En otros sistemas, los segmentos 523 se denominan grupos de mosaicos que contienen los mosaicos. Los segmentos 523 y/o grupos de mosaicos de mosaicos hacen referencia a un encabezado de segmento 515. Los segmentos 523 además se dividen en unidades de árbol de codificación (CTU). Las CTU se dividen además en bloques de codificación basados en árboles de codificación. A continuación, los bloques de codificación se pueden codificar/decodificar según los mecanismos de predicción.

La imagen 521 y/o el segmento 523 pueden referirse, directa o indirectamente, a un APS de ALF 512, el APS de lista de escalado 513 y/o el APS de LMCS 514 que contienen parámetros relevantes. Por ejemplo, un segmento 523 puede referirse a un encabezado de segmento 515. Además, una imagen 521 puede referirse a un correspondiente encabezado de imagen. El encabezado de segmento 515 y/o encabezado de imagen puede referirse al APS de ALF 512, el APS de lista de escalado 513, y/o el APS de LMCS 514 que contiene parámetros usados en codificación el segmento 523 y/o imagen 521 relevantes. De esta manera, un decodificador puede obtener los parámetros de herramienta de codificación que son relevantes para un segmento 523 y/o imagen 521 según referencias de encabezado relacionadas con el correspondiente segmento 523 y/o imagen 521.

El flujo de bits 500 se codifica en unidades de NAL de capa de codificación de vídeo (VCL) 535 y unidades de NAL no VCL 531. Una unidad de NAL es una unidad de datos codificados dimensionadas a colocar como carga útil para un único paquete para transmisión por un red. Una unidad de NAL de VCL 535 es una unidad de NAL que contiene datos de vídeo codificado. Por ejemplo, cada unidad de NAL de VCL 535 puede contener un segmento 523 y/o grupo de mosaicos de datos, CTU, y/o bloques de codificación. Una unidad de NAL no VCL 531 es una unidad de NAL que contiene sintaxis de soporte, pero no contiene datos de vídeo codificados. Por ejemplo, una unidad de NAL no VCL 531 puede contener el SPS 510, un PPS 511, un APS, un encabezado de segmento 515, etc. Como tal, el decodificador recibe el flujo de bits 500 en unidades de NAL de VCL 535 y unidades de NAL no VCL 531 discretas. Una unidad de acceso es un grupo de unidades de NAL de VCL 535 y/o unidades de NAL no VCL 531 que incluyen datos suficientes para codificar una única imagen 521.

En algunos ejemplos, el APS de ALF 512, el APS de lista de escalado 513, y el APS de LMCS 514 se asignan cada uno a un tipo de unidad de NAL no VCL separado 531. En tal caso, el APS de ALF 512, el APS de lista de escalado 513 y el APS de LMCS 514 se incluyen en una unidad de NAL de APS de ALF 532, una unidad de NAL de APS de lista de escalado 533, y una unidad de NAL de APS de LMCS 534, respectivamente. Por consiguiente, una unidad de NAL de APS de ALF 532 contiene parámetros de ALF que permanecen vigentes hasta que se recibe otra unidad de NAL de APS de ALF 532. Además, una unidad de NAL de APS de lista de escalado 533 contiene parámetros de lista de escalado que permanecen vigentes hasta que se recibe otra unidad de NAL de APS de lista de escalado 533. Adicionalmente, una unidad de NAL de APS de LMCS 534 contiene parámetros de LMCS que permanecen vigentes hasta que se recibe otra unidad de NAL de APS de LMCS 534. De esta manera, no hay necesidad de emitir un nuevo APS cada vez que cambia un parámetro de APS. Por ejemplo, un cambio en parámetros de LMCS tiene como resultado un APS de LMCS 514 adicional, pero no tiene como resultado un APS de ALF 512 o APS de lista de escalado 513 adicional. Por consiguiente, al separar APS en diferentes tipos de unidad de NAL en función del tipo de parámetro, se evita la señalización redundante de parámetros no relacionado. Como tal, separar APS en diferentes tipos de unidad de NAL aumenta la eficiencia de codificación, y por tanto disminuye el uso de procesador, memoria y/o recursos de red en el codificador y el decodificador.

Además, un segmento 523 y/o una imagen 521 pueden hacer referencia directa o indirectamente un APS de ALF 512, unidad de NAL de APS de ALF 532, APS de lista de escalado 513, unidad de NAL de APS de lista de escalado 533, APS de LMCS 514, y/o una unidad de NAL de APS de LMCS 534 que contiene parámetros de herramienta de codificación empleados para codificar el segmento 523 y/o una imagen 521. Por ejemplo, cada APS puede contener un ID de APS 542 y un tipo de parámetro 541. El ID de APS 542 es un valor (p. ej., un número) que identifica el correspondiente APS. El ID de APS 542 puede contener un número de bits predefinido. Por lo tanto, el ID de APS 542 puede aumentar según una secuencia predefinida (p. ej., aumentar en uno) y puede restablecerse en un valor mínimo (p. ej., cero) una vez que la secuencia alcanza el final de un intervalo predefinido. El tipo de parámetro 541 indica el tipo de parámetros contenidos en el APS (p. ej., ALF, lista de escalado, y/o LMCS). Por ejemplo, el tipo de parámetro 541 puede incluir un código de tipo de parámetro de APS (`aps_params_type`) establecido a un valor predefinido que indica un tipo de parámetros incluidos en cada APS. Como tal, el tipo de parámetro 541 se puede usar para distinguir entre un APS de ALF 512, un APS de lista de escalado 513, y un APS de LMCS 514. En algunos ejemplos, el APS de ALF 512, el APS de lista de escalado 513 y el APS de LMCS 514 pueden identificarse cada uno de manera única por una combinación de un tipo de parámetro 541 y un ID de APS 542. Por ejemplo, cada tipo de APS puede incluir un espacio de valores separado para correspondientes ID de APS 542. Por tanto, cada tipo de APS puede incluir un ID de APS 542 que aumenta en secuencia en función de un APS anterior del mismo tipo. Sin embargo, el ID de APS 542 para un primer tipo de APS puede no relacionarse con un ID de APS 542 para un APS anterior de un segundo tipo de APS diferente. Como tal, ID de APS 542 para tipos diferentes de APS pueden incluir espacios de valores que se superponen. Por ejemplo, un APS de un primer tipo (p. ej. un APS de ALF) puede incluir el mismo ID de APS 542 como APS de un segundo tipo (p. ej. un APS de LMCS) en algunos casos. Al permitir que cada tipo de APS incluya un espacio de valores diferente, el códec no necesita comprobar conflictos de ID de APS 542 en tipos de APS. Además, al permitir que los espacios de valor se superpongan, el códec puede evitar emplear valores de ID de APS 542 más grandes, lo que tiene como resultado ahorros de bits. Como tal, emplear espacios de valor superpuestos separados para los ID de APS 542 de diferentes tipos de APS aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador. Como se señala anteriormente, el ID de APS 542 se puede extender en un intervalo predefinido. En algunos ejemplos, el intervalo predefinido del ID de APS 542 puede variar dependiendo del tipo de APS indicado por el tipo de parámetro 541. Esto puede permitir asignar un número diferente de bits a tipos diferentes de APS dependiendo de lo a menudo que cambien generalmente parámetros de tipos diferentes. Por ejemplo, los ID de APS 542 del APS de ALF 512

pueden tener un intervalo de cero a siete, los ID de APS 542 del APS de lista de escalado 513 pueden tener un intervalo de cero a siete, y los ID de APS 542 del APS de LMCS 514 puede tener un intervalo de cero a tres.

En otro ejemplo, parámetros de LMCS se incluyen en el APS de LMCS 514. Algunos sistemas incluyen parámetros de LMCS en un encabezado de segmento 515. Sin embargo, parámetros de LMCS/remodelador pueden cambiar aproximadamente una vez por segundo. Las secuencias de vídeo pueden visualizar de treinta a sesenta imágenes 521 por segundo. Como tal, parámetros de LMCS pueden no cambiar para treinta a sesenta fotogramas. Incluir parámetros de LMCS en un APS de LMCS 514 reduce significativamente la codificación redundante de parámetros de LMCS. En algunos ejemplos, un encabezado de segmento 515 y/o un encabezado de imagen asociados con un segmento 523 y/o imagen 521, respectivamente pueden hacer referencia el APS de LMCS relevante 514. El segmento 523 y/o imagen 521 hace referencia entonces al encabezado de segmento 515 y/o un encabezado de imagen. Esto permite al decodificador obtener los parámetros de LMCS para el segmento 523 y/o imagen 521 asociados. De esta manera, los parámetros de LMCS únicamente se codifican cuando cambian los parámetros de LMCS para segmento 523 y/o imagen 521. Por consiguiente, emplear un APS de LMCS 514 para codificar parámetros LMCS aumenta la eficiencia de codificación y, por tanto, reduce el uso de recursos de red, recursos de memoria y/o recursos de procesamiento en el codificador y el decodificador. Como no se emplea LMCS para todos los vídeos, el SPS 510 puede incluir un indicador habilitado por LMCS 543. El indicador habilitado por LMCS 543 se puede establecer para indicar que LMCS está habilitado para una secuencia de vídeo codificada. Como tal, el decodificador puede obtener los parámetros de LMCS del APS de LMCS 514 en función del indicador habilitado por LMCS 543 cuando se establece el indicador habilitado por LMCS 543 (p. ej., a uno). Además, el decodificador puede no intentar obtener parámetros de LMCS cuando el indicador habilitado por LMCS 543 no está establecido (p. ej., a cero).

La FIGURA 6 es un diagrama esquemático que ilustra un mecanismo de ejemplo 600 para asignar ID de APS 642 a tipos diferentes de APS en diferentes espacios de valores. Por ejemplo, el mecanismo 600 se puede aplicar a un flujo de bits 500 para asignar ID de APS 542 a APS de ALF 512, APS de lista de escalado 513 y/o APS de LMCS 514. Además, el mecanismo 600 puede aplicarse a un códec 200, un codificador 300 y/o un decodificador 400 cuando se codifica un vídeo según el método 100.

El mecanismo 600 asigna ID de APS 642 a APS de ALF 612, APS de lista de escalado 613 y APS de LMCS 614, que puede ser sustancialmente similar a ID de APS 542, APS de ALF 512, APS de lista de escalado 513 y APS de LMCS 514, respectivamente. Como se señala anteriormente, el ID de APS 642 puede asignarse en secuencia en una pluralidad de espacios de valores diferentes donde cada espacio de valores es específico para el tipo de APS. Además, cada espacio de valores se puede extender en un intervalo diferente que es específico para el tipo de APS. En el ejemplo mostrado, el intervalo del espacio de valores del ID de APS 642 para el APS de ALF 612 es desde cero a siete (p. ej., tres bits). Además, el intervalo del espacio de valores del ID de APS 642 para el APS de lista de escalado 613 es desde cero a siete (p. ej., tres bits). También, el intervalo del espacio de valores del ID de APS 642 para el APS de LMCS 611 es desde cero a tres (p. ej., dos bits). Cuando un ID de APS 642 alcanza la extremo del intervalo de espacio de valores, el ID de APS 642 del siguiente APS del correspondiente tipo vuelve al inicio del intervalo (p. ej., cero). Cuando un nuevo APS recibe el mismo ID de APS 642 como APS anterior del mismo tipo, el APS anterior ya no está activo y puede ya no ser referenciado. Como tal, el intervalo de un espacio de valores se puede expandir para permitir que más APS de un tipo sean referenciados activamente. Además, el intervalo de un espacio de valores se puede reducir para aumento de eficiencia de codificación, pero tal reducción también reduce el número de APS del tipo correspondiente que puede permanecer simultáneamente activo y disponible para referenciar.

En el ejemplo mostrado, cada APS de ALF 612, APS de lista de escalado 613 y APS de LMCS 614 se referencian por una combinación de ID de APS 642 y tipo de APS. Por ejemplo, el APS de ALF 612, el APS de LMCS 614 y el APS de lista de escalado 613 reciben cada uno un ID de APS 642 de cero. Cuando se recibe un nuevo APS de ALF 612, el ID de APS 642 se incrementa desde el valor usado para el APS anterior de ALF 612. La misma secuencia se aplica para el APS de lista de escalado 613 y APS de LMCS 614. Por consiguiente, cada ID de APS 642 se relaciona con el ID de APS 642 para APS anteriores del mismo tipo. Sin embargo, el ID de APS 642 no se relaciona con un ID de APS 642 de APS anteriores de otros tipos. En este ejemplo, el ID de APS 642 del APS de ALF 612 aumenta incrementalmente desde cero a siete y entonces regresa a cero antes de seguir aumentando. Además, el ID de APS 642 del APS de lista de escalado 613 aumenta incrementalmente desde cero a siete y entonces regresa a cero antes de seguir aumentando. También, el ID de APS 642 del APS de LMCS 611 aumenta incrementalmente desde cero a tres y entonces regresa a cero antes de seguir aumentando. Como se muestra, tales espacios de valores se superponen ya que diferentes APS de diferentes tipos de APS pueden compartir el mismo ID de APS 642 en el mismo punto en la secuencia de vídeo. También cabe señalar que el mecanismo 600 únicamente representa el APS. En un flujo de bits, el APS representado se intercalaría entre otras unidades de NAL de VCL y no VCL, tales como un SPS, PPS, encabezado de segmento, encabezado de imagen, segmentos, etc.

Como tal, la presente invención incluye mejoras al diseño del APS así como algunas mejoras para señalar de los parámetros de remodelador/LMCS. El APS se diseña para señalar de información que se puede compartir por múltiples imágenes y puede incluir muchas variaciones. Los parámetros de remodelador/LMCS se emplean para

una herramienta de codificación de vídeo adaptativo en bucle de remodelador/LMCS . Los anteriores mecanismos se pueden implementar de la siguiente manera. A fin de resolver los problemas enumerados en esta memoria, esto incluye varios aspectos que se pueden emplear individualmente y/o en combinación.

El APS descrito se modifica de manera que se pueden usar múltiples APS para llevar tipos diferentes de parámetros. Cada unidad de NAL de APS se usa para llevar únicamente un tipo de parámetros. En consecuencia, dos unidades de NAL de APS se codifican cuando dos tipos de información se llevan para un grupo de mosaicos/segmento particular (p. ej., uno para cada tipo de información). El APS puede incluir un campo de tipo de parámetros de APS en la sintaxis de APS. Únicamente parámetros del tipo indicado por el campo de tipo de parámetros de APS se puede incluir en la unidad de NAL de APS.

En algunos ejemplos, tipos diferentes de parámetros de APS se indican por diferentes tipos de unidad de NAL. Por ejemplo, dos tipos diferentes de unidad de NAL se usan para APS. Los dos tipos de APS pueden denominarse APS de ALF y APS de remodelador, respectivamente. En otro ejemplo, un tipo de parámetros de herramienta que se llevan en una unidad de NAL de APS se especifica en el encabezado de unidad de NAL. En VVC, un encabezado de unidad de NAL tiene bits reservados (p. ej., siete bits denotados que `nuh_reserved_zero_7bits`). En algunos ejemplos, algunos de estos bits (p. ej., tres bits de los siete bits) se pueden usar para especificar un campo de tipo de parámetro de APS. En algunos ejemplos, APS de un tipo particular pueden compartir el mismo espacio de valores para los ID de APS. Entretanto, tipos diferentes de APS usan diferentes espacios de valores del ID de APS. Por consiguiente, dos APS de tipos diferentes pueden coexistir y tener el mismo valor de ID de APS en el mismo instante. Además, la combinación de ID de APS y tipo de parámetros de APS se puede emplear para identificar un APS de otros APS.

El ID de APS se puede incluir en la sintaxis de encabezado de grupo de mosaicos cuando una correspondiente herramienta de codificación se habilita para el grupo de mosaicos. De otro modo el ID de APS del correspondiente tipo puede no incluirse en el encabezado de grupo de mosaicos. Por ejemplo, cuando ALF se habilita para un grupo de mosaicos, el ID de APS del APS de ALF se incluye en el encabezado de grupo de mosaicos. Por ejemplo, esto se puede conseguir estableciendo el campo de tipo de parámetros de APS para indicar el tipo de ALF. Por consiguiente, cuando ALF no se habilita para el grupo de mosaicos, el ID de APS del APS de ALF no se incluye en el encabezado de grupo de mosaicos. Además, cuando una herramienta de codificación de remodelador se habilita para un grupo de mosaicos, el ID de APS del APS de remodelador se incluye en el encabezado de grupo de mosaicos. Por ejemplo, esto se puede conseguir estableciendo el campo de tipo de parámetros de APS para indicar el tipo de remodelador. Además, cuando la herramienta de codificación de remodelador no se habilita para el grupo de mosaicos, el ID de APS del APS de remodelador puede no incluirse en el encabezado de grupo de mosaicos.

En algunos ejemplos, la presencia de información de tipo de parámetros de APS en un APS puede ser acondicionada por el uso de herramientas de codificación asociadas con los parámetros. Cuando únicamente se habilita una herramienta de codificación relacionada con APS para un flujo de bits (p. ej., LMCS, ALF, o lista de escalado), puede no haber presente información de tipo de parámetros de APS y en cambio puede inferirse. Por ejemplo, cuando un APS puede contener parámetros para herramientas de codificación de ALF y remodelador pero únicamente se habilita ALF (p. ej., como se especifica por un indicador en el SPS) y no se habilita remodelador, (p. ej., como se especifica por un indicador en el SPS), el tipo de parámetro de APS puede no señalarse y se puede inferir como que es igual a parámetros de ALF.

En otro ejemplo, información de tipo de parámetro de APS puede inferirse del valor de ID de APS. Por ejemplo, un intervalo predefinido de valor de ID de APS se puede asociar con un correspondiente tipo de parámetro de APS. Este aspecto se puede implementar de la siguiente manera. En lugar de asignar X bits para señalar un ID de APS e Y bits para señalar tipo de parámetros de APS, se pueden asignar X + Y bits para señalar el ID de APS. Los diferentes intervalos del valor del ID de APS pueden entonces especificarse para indicar tipos diferentes de tipos de parámetro de APS. Por ejemplo, en lugar de usar cinco bits para señalar un ID de APS y tres bits para señalar un tipo de parámetros de APS, se pueden asignar ocho bits para señalar un ID de APS (p. ej., sin aumentar los costes de bits). El intervalo de valor de ID de APS de cero a sesenta tres indica que el APS contiene parámetros para ALF, el intervalo de valor de ID de sesenta cuatro a noventa cinco indica que el APS contiene parámetros para remodelador, y de noventa seis a doscientos cincuenta y cinco se puede reservar para otros tipos de parámetros, tales como lista de escalado. En otro ejemplo, el intervalo de valor de ID de APS de cero a treinta y uno indica que el APS contiene parámetros para ALF, el intervalo de valor de ID de treinta dos a cuarenta y siete indica que el APS contiene parámetros para remodelador, y de cuarenta y ocho a doscientos cincuenta y cinco se puede reservar para otros tipos de parámetros, tales como lista de escalado. La ventaja de este planteamiento es que el intervalo de ID de APS se puede asignar dependiendo de la frecuencia de los cambios de parámetro de cada herramienta. Por ejemplo, puede esperarse que parámetros de ALF cambien más frecuentemente que parámetros de remodelador. En tal caso, se puede emplear un intervalo de ID de APS más grande para indicar un APS es un APS que contiene parámetro de ALF.

En un primer ejemplo, uno o más de los aspectos anteriores pueden implementarse de la siguiente manera. Un APS de ALF se puede definir como un APS que tiene `aps_params_type` igual a `ALF_APS`. Un APS de remodelador

(o APS de LMCS) puede definirse que un APS que tiene `aps_params_type` igual a `MAP_APS`. Un ejemplo de sintaxis y semántica de SPS son la siguiente manera.

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>sps_resampler_enabled_flag</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	

- 5 El `sps_resampler_enabled_flag` se establece igual a uno para especificar que se usa remodelador en la secuencia de vídeo codificada (CVS). El `sps_resampler_enabled_flag` se establece igual a cero para especificar que no se usa remodelador en el CVS.

Un ejemplo de sintaxis y semántica de APS son de la siguiente manera.

10

<code>adaptation_parameter_set_rbsp() {</code>	Descriptor
<code>adaptation_parameter_set_id</code>	<code>u(5)</code>
<code>aps_params_type</code>	<code>u(3)</code>
<code>if(aps_params_type == ALF_APS) // 0</code>	
<code>alf_data()</code>	
<code>else if(aps_params_type == MAP_APS) // 1</code>	
<code>resampler_data()</code>	
<code>aps_extension_flag</code>	<code>u(1)</code>
<code>if(aps_extension_flag)</code>	
<code>while(more_rbsp_data())</code>	
<code>aps_extension_data_flag</code>	<code>u(1)</code>
<code>rbp_trailing_bits()</code>	
<code>}</code>	

El `aps_params_type` especifica el tipo de parámetros de APS llevados en el APS como se especifica en el siguiente tabla.

15

Tabla 1 - códigos de tipo de parámetros de APS y tipos de parámetros de APS

<code>aps_params_type</code>	Nombre de <code>aps_params_type</code>	Tipo de parámetros APS
0	<code>ALF_APS</code>	Parámetros de ALF
1	<code>MAP_APS</code>	Parámetros de correlación en bucle (es decir, remodelador)
2..7	Reservado	Reservado

- 20 Un ejemplo de sintaxis y semántica de grupo de mosaicos es de la siguiente manera.

<code>tile_group_header() {</code>	Descriptor
<code>...</code>	
<code>if(sps_alf_enabled_flag) {</code>	
<code>tile_group_alf_enabled_flag</code>	<code>u(1)</code>
<code>if(tile_group_alf_enabled_flag)</code>	

tile_group_header() {	Descriptor
tile_group_alf_aps_id	u(5)
}	
...	
if(sps_reshaper_enabled_flag) {	
tile_group_reshaper_enable_flag	u(1)
tile_group_reshaper_aps_id	u(5)
if(tile_group_reshaper_enable_flag && (! qtbtt_dual_tree_intra_flag &&	
tile_group_type = I)))	
tile_group_reshaper_chroma_residual_scale_flag	u(1)
}	
...	
if(NumTilesInCurrTileGroup > 1) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumTilesInCurrTileGroup - 1; i++)	
entry_point_offset_minus1[i]	u(v)
}	
byte_alignment()	
}	

- El tile_group_alf_aps_id especifica el adaptation_parameter_set_id del APS de ALF al que se refiere el grupo de mosaicos. El TemporalId de la unidad de NAL de APS de ALF que tiene adaptation_parameter_set_id igual a tile_group_alf_aps_id será menor o igual al TemporalId de la unidad de NAL de grupo de mosaicos codificados. Cuando múltiples APS de ALF con el mismo valor de adaptation_parameter_set_id son referidos por dos o más grupos de mosaicos de la misma imagen, los múltiples APS de ALF con el mismo valor de adaptation_parameter_set_id tendrán el mismo contenido.
- El tile_group_reshaper_enabled_flag se establece igual a uno para especificar que se habilita remodelador para el grupo de mosaicos actual. El tile_group_reshaper_enabled_flag se establece igual a cero para especificar que el remodelador no se habilita para el grupo de mosaicos actual. Cuando tile_group_reshaper_enable_flag no está presente, se infiere que el indicador es igual a cero. El tile_group_reshaper_aps_id especifica el adaptation_parameter_set_id del APS de remodelador al que se refiere el grupo de mosaicos. El TemporalId de la unidad de NAL de APS de remodelador que tiene adaptation_parameter_set_id igual a tile_group_reshaper_aps_id será menor o igual que el TemporalId de la unidad de NAL de grupo de mosaicos codificados. Cuando múltiples APS de remodelador con el mismo valor de adaptation_parameter_set_id son referidos por dos o más grupos de mosaicos de la misma imagen, los múltiples APS de remodelador con el mismo valor de adaptation_parameter_set_id tendrán el mismo contenido. El tile_group_reshaper_chroma_residual_scale_flag se establece igual uno para especificar que escalado residual de croma se habilita para el grupo de mosaicos actual. El tile_group_reshaper_chroma_residual_scale_flag se establece igual a cero para especificar que escalado residual de croma no se habilita para el grupo de mosaicos actual. Cuando tile_group_reshaper_chroma_residual_scale_flag no está presente, se infiere que el indicador es igual a cero.
- Un ejemplo de sintaxis y semántica de datos de remodelador es de la siguiente manera.

reshaper_data() {	Descriptor
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for(i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshaper_model_bin_delta_abs_CW[i]	u(v)

reshaper_data() {	Descriptor
if(reshaper_model_bin_delta_abs_CW[i]) > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

- El reshaper_model_min_bin_idx especifica el índice mínimo de contenedor (o pedazo) a usar en el proceso de construcción de remodelador. El valor de reshaper_model_min_bin_idx estará en el intervalo de cero a MaxBinIdx, inclusive. El valor de MaxBinIdx será igual a quince. El reshaper_model_delta_max_bin_idx especifica el máximo índice de contenedor permitido (o pedazo) MaxBinIdx menos el máximo índice de contenedor a usar en el proceso de construcción de remodelador. El valor de reshaper_model_max_bin_idx se establece igual a MaxBinIdx - reshaper_model_delta_max_bin_idx. El reshaper_model_bin_delta_abs_cw_prec_minus1 plus 1 especifica el número de bits usado para la representación del elemento de sintaxis reshaper_model_bin_delta_abs_CW[i].
- 5 El reshaper_model_bin_delta_abs_CW[i] especifica el valor absoluto de palabra clave delta para la i-ésimo contenedor. El elemento de sintaxis reshaper_model_bin_delta_abs_CW[i] se representa por
- 10 reshaper_model_bin_delta_abs_cw_prec_minus1 + 1 bits. El reshaper_model_bin_delta_sign_CW_flag[i] especifica el signo de reshaper_model_bin_delta_abs_CW[i].
- 15 En un segundo ejemplo, uno o más de los aspectos anteriores puede implementarse de la siguiente manera. Un ejemplo sintaxis y semántica de SPS es de la siguiente manera.

seq_parameter_set_rbsp() {	Descriptor
...	
sps_alf_enabled_flag	u(1)
...	
sps_reshaper_enabled_flag	u(1)
...	
}	

- 20 La variable ALFEnabled y ReshaperEnabled se establecen de la siguiente manera. ALFEnabled = sps_alf_enabled_flag y ReshaperEnabled = sps_reshaper_enabled_flag.

Un ejemplo de sintaxis y semántica de APS son de la siguiente manera.

adaptation_parameter_set_rbsp() {	Descriptor
adaptation_parameter_set_id	u(5)
if(ALFEnabled && ReshaperEnabled)	
aps_params_type	u(3)
if(aps_params_type == ALF_APS) // 0	
alf_data()	
else if (aps_params_type == MAP_APS) // 1	
reshaper_data()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

El `aps_params_type` especifica el tipo de parámetros de APS llevados en el APS como se especifica en el siguiente tabla.

5

Tabla 2 - códigos de tipo de parámetros de APS y tipos de parámetros de APS

aps_params_type	Nombre de aps_params_type	Tipo de parámetros APS
0	ALF_APS	Parámetros de ALF
1	MAP_APS	Parámetros de correlación en bucle (es decir, remodelador)
2..7	Reservado	Reservado

10 Cuando no está presente, el valor de `aps_params_type` se infiere de la siguiente manera. Si `ALFEnabled`, `aps_params_type` se establece para ser igual a cero. Si no, `aps_params_type` se establece para ser igual a uno.

En un segundo ejemplo, uno o más de los aspectos anteriores puede implementarse de la siguiente manera. Un ejemplo sintaxis y semántica de SPS es de la siguiente manera.

15

<code>adaptation_parameter_set_rbsp() {</code>	Descriptor
<code>adaptation_parameter_set_id</code>	<code>u(8)</code>
<code>if(ApsParamsType == ALF_APS) // 0</code>	
<code>alf_data()</code>	
<code>else if (ApsParamsType == MAP_APS) // 1</code>	
<code>reshaper_data()</code>	
<code>aps_extension_flag</code>	<code>u(1)</code>
<code>if(aps_extension_flag)</code>	
<code>while(more_rbsp_data())</code>	
<code>aps_extension_data_flag</code>	<code>u(1)</code>
<code>rbp_trailing_bits()</code>	
<code>}</code>	

El `adaptation_parameter_set_id` proporciona un identificador para el APS para referencia por otros elementos de sintaxis. Los APS se pueden compartir por imágenes y pueden ser diferentes en diferentes grupos de mosaicos dentro de una imagen. El valor y la descripción de la variable `APSPParamsType` se definen en el siguiente tabla.

20

Tabla 3 - códigos de tipo de parámetros de APS y tipos de parámetros de APS

Intervalo de ID de APS	ApsParamsType	Tipo de parámetros APS
0~63	0: ALF_APS	Parámetros de ALF
64~95	1: MAP_APS	Parámetros de correlación en bucle (es decir, remodelador)
96..255	Reservado	Reservado

25

Un ejemplo de semántica de encabezado grupo de mosaicos es de la siguiente manera. El `tile_group_alf_aps_id` especifica el `adaptation_parameter_set_id` del APS de ALF al que se refiere el grupo de mosaicos. El `TemporalId` de la unidad de NAL de APS de ALF que tiene `adaptation_parameter_set_id` igual a `tile_group_alf_aps_id` será menor o igual al `TemporalId` de la unidad de NAL de grupo de mosaicos codificados. El valor de `tile_group_alf_aps_id` estará en el intervalo de cero a sesenta y tres, inclusive.

30

Quando múltiples APS de ALF con el mismo valor de `adaptation_parameter_set_id` son referidos por dos o más grupos de mosaicos de la misma imagen, los múltiples APS de ALF con el mismo valor de `adaptation_parameter_set_id` tendrán el mismo contenido. El `tile_group_reshaper_aps_id` especifica el `adaptation_parameter_set_id` del APS de remodelador al que se refiere el grupo de mosaicos. El TemporalId de la unidad de NAL de APS de remodelador que tiene `adaptation_parameter_set_id` igual a `tile_group_reshape_aps_id` será menor o igual al TemporalId de la unidad de NAL de grupo de mosaicos codificado. El valor de `tile_group_reshaper_aps_id` estará en el intervalo de sesenta y cuatro a noventa y cinco, inclusive. Cuando múltiples APS de remodelador con el mismo valor de `adaptation_parameter_set_id` son referidos por dos o más grupos de mosaicos de la misma imagen, los múltiples APS de remodelador con el mismo valor de `adaptation_parameter_set_id` tendrán el mismo contenido.

La FIGURA 7 es un diagrama esquemático de un ejemplo de dispositivo de codificado de vídeo 700. El dispositivo de codificado de vídeo 700 es adecuado para implementar las realizaciones/ejemplos descritos en esta memoria. El dispositivo de codificado de vídeo 700 comprende puertos aguas abajo 720, puertos aguas arriba 750 y/o unidades de transceptor (Tx/Rx) 710, que incluyen transmisores y/o receptores para comunicar datos aguas arriba y/o aguas abajo a través de una red. El dispositivo de codificado de vídeo 700 también incluye un procesador 730 que incluye una unidad lógica y/o unidad de procesamiento central (CPU) para procesar los datos y una memoria 732 para almacenar los datos. El dispositivo de codificado de vídeo 700 también puede comprender componentes eléctricos, ópticos o eléctricos (OE), componentes eléctricos a ópticos (EO) y/o componentes de comunicación inalámbrica acoplados a los puertos aguas arriba 750 y/o aguas abajo 720 para la comunicación de datos a través de redes de comunicación eléctricas, ópticas o inalámbricas. El dispositivo de codificado de vídeo 700 también puede incluir dispositivos de entrada y/o salida (E/S) 760 para comunicar datos hacia y desde un usuario. Los dispositivos de E/S 760 pueden incluir dispositivos de salida tales como un visualizador para visualizar datos de vídeo, altavoces para emitir datos de audio, etc. Los dispositivos de E/S 760 también pueden incluir dispositivos de entrada, tales como un teclado, ratón, bola de seguimiento, etc. y/o interfaces correspondientes para interactuar con dichos dispositivos de salida.

El procesador 730 se implementa por hardware y software. El procesador 730 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador de múltiples núcleos), matrices de puertas programables en campo (FPGA), circuitos integrados específicos de aplicación (ASIC) y procesadores de señales digitales (DSP). El procesador 730 está en comunicación con los puertos aguas abajo 720, Tx/Rx 710, los puertos aguas arriba 750 y la memoria 732. El procesador 730 comprende un módulo de codificado 714. El módulo de codificado 714 implementa las realizaciones divulgadas descritas en esta memoria, tales como los métodos 100, 800 y 900, que pueden emplear un flujo de bits 500 y/o mecanismo 600. El módulo de codificado 714 también puede implementar cualquier otro método/mecanismo descrito en esta memoria. Además, el módulo de codificado 714 puede implementar un sistema de códec 200, un codificador 300 y/o un decodificador 400. Por ejemplo, el módulo de codificado 714 puede codificar/decodificar imágenes en un flujo de bits y codificar/decodificar parámetros asociados con segmentos de las imágenes en una pluralidad de APS. En algunos ejemplos, tipos diferentes de parámetros se pueden codificar en tipos diferentes de APS. Además, tipos diferentes de APS se pueden incluir en diferentes tipos de unidad de NAL. Tales tipos de APS pueden incluir APS de ALF, APS de lista de escalad y/o APS de LMCS. El APS puede incluir ID de APS. Los ID de APS de tipos diferentes de APS pueden aumentar en secuencia en diferentes espacios de valores. También, segmentos y/o imágenes pueden hacer referencia a correspondientes encabezados de segmentos y/o encabezados de imagen. Tales encabezados pueden entonces hacer referencia a APS que contienen herramientas de codificación relevantes. Tales APS se pueden referenciar de manera única por ID de APS y tipo de APS. Tales ejemplos reducen la señalización redundante de parámetros de herramienta de codificación y/o reducen el uso de bits para identificadores. Como tal, el módulo de codificado 714 hace que el dispositivo de codificado de vídeo 700 proporcione funcionalidad adicional y/o eficiencia de codificación al codificar datos de vídeo. Como tal, el módulo de codificado 714 mejora la funcionalidad del dispositivo de codificado de vídeo 700 así como también que aborda problemas que son específicos de las técnicas de codificación de vídeo. Además, el módulo de codificado 714 efectúa una transformación del dispositivo de codificado de vídeo 700 a un estado diferente. Alternativamente, el módulo de codificado 714 puede implementarse como instrucciones almacenadas en la memoria 732 y ejecutadas por el procesador 730 (por ejemplo, como un producto de programa informático almacenado en un medio no transitorio).

La memoria 732 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 direccionable de contenido ternario (TCAM), memoria de acceso aleatorio estática (SRAM), etc. La memoria 732 puede usarse como un dispositivo de almacenamiento de datos de desbordamiento, para almacenar programas cuando dichos programas se seleccionan para su ejecución, y para almacenar instrucciones y datos que se leen durante la ejecución del programa.

La FIGURA 8 es un diagrama de flujo de un método de ejemplo 800 de codificar una secuencia de vídeo hasta un flujo de bits, tal como un flujo de bits 500, empleando una pluralidad de tipos de APS, tales como APS ALF 512, APS de lista de escalado 513, y/o APS de LMCS 514. El método 800 puede ser empleado por un codificador, como un sistema de códec 200, un codificador 300, y/o un dispositivo de codificado de vídeo 700 al realizar el método

100. El método 800 también puede asignar ID de APS a APS de tipos diferentes empleando diferentes espacios de valores según el mecanismo 600.

El método 800 puede comenzar cuando un codificador recibe una secuencia de vídeo que incluye una pluralidad de imágenes y determina codificar esa secuencia de vídeo en un flujo de bits, por ejemplo en función de la entrada del usuario. La secuencia de vídeo se particiona en fotografías/imágenes/fotogramas para su posterior particionado antes de la codificación. En la etapa 801, un segmento se codifica hasta un flujo de bits como parte de una secuencia de vídeo codificada. El segmento puede codificarse como parte de una imagen. Además, el segmento puede codificarse al codificar CTU y/o CU contenidos en el segmento. Tales CU se pueden codificar según intrapredicción y/o interpredicción. Por ejemplo, el codificador puede codificar las CU del segmento. El codificador puede entonces emplear un decodificador de referencia hipotético (HRD) para decodificar el segmento codificado y filtrar el segmento decodificado para aumentar la calidad de salida del segmento.

En la etapa 803, el codificador determina una pluralidad de tipos de parámetros empleados para codificar el segmento. Tales parámetros pueden incluir parámetros empleados por filtros aplicados por el HRD para aumentar la calidad del segmento codificado. Estos parámetros pueden incluir parámetros de ALF, parámetros de lista de escalado, y parámetros de LMCS. Determinar estos parámetros permite al codificador determinar los parámetros que deben ser empleados para aplicación al segmento en el decodificador.

En la etapa 805, la pluralidad de tipos de parámetros se codifica en una pluralidad de APS. Específicamente, la pluralidad de tipos de parámetros se separan por tipo y se incluyen en APS de tipos correspondientes. Por consiguiente, la pluralidad de tipos de parámetros se incluye en una pluralidad de tipos de APS. Por ejemplo, la pluralidad de tipos de APS puede incluir un tipo de ALF APS que contiene parámetros de ALF, un tipo de APS de lista de escalado que contiene parámetros de lista de escalado, y un APS de tipo de LMCS que contiene parámetros de LMCS.

En la etapa 807, ID de APS se asignan a cada uno de los APS. Esto incluye asignar ID de APS para los tipos de APS en secuencia en una pluralidad de diferentes espacios de valores. Por consiguiente, un ID de APS se asigna a cada uno de los APS de manera que cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS. Por ejemplo, cada uno de los diferentes espacios de valores se puede extender en un intervalo predefinido. Además, los intervalos predefinidos se pueden determinar en función de los tipos de APS como se especifica por un tipo de parámetro del correspondiente APS. Como tal, cada APS puede incluir un ID de APS seleccionado de un intervalo predefinido. Tales intervalos predefinidos pueden superponerse, y por tanto la pluralidad de diferentes espacios de valores también pueden superponerse. Además, los ID de APS pueden asignarse según diferentes secuencias en función de los tipos de APS. Por ejemplo, un APS actual puede incluir un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual. El ID de APS actual se relaciona secuencialmente con un ID de APS anterior asociado con un APS anterior del mismo tipo que el APS actual. Además, el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual ya que la secuencias para los tipos diferentes de APS son diferentes.

En la etapa 809, los ID de APS se codifican en la pluralidad de APS. Un tipo de APS también puede codificarse en cada APS. Por ejemplo, cada APS puede incluir un código de tipo de parámetro de APS (p. ej., `aps_params_type`) establecido a un valor predefinido que indica un tipo de parámetros incluidos en el correspondiente APS. Cada APS puede entonces identificarse por una combinación de un tipo de APS actual y un ID de APS actual. El flujo de bits se almacena en memoria. Bajo petición, el flujo de bits puede entonces comunicarse hacia un decodificador, por ejemplo por medio de un transmisor.

La FIGURA 9 es un diagrama de flujo de un método de ejemplo 900 de decodificar una secuencia de vídeo desde un flujo de bits, tal como un flujo de bits 500, empleando una pluralidad de tipos de APS, tales como APS ALF 512, APS de lista de escalado 513, y/o APS de LMCS 514. El método 900 puede ser empleado por un decodificador, como un sistema de códec 200, un decodificador 400 y/o un dispositivo de codificado de vídeo 700 al realizar el método 100. El método 900 también puede hacer referencia a APS en función de ID de APS asignados según el mecanismo 600 donde APS de tipos diferentes emplean ID de APS asignados según diferentes espacios de valores.

El método 900 puede comenzar cuando un decodificador comienza a recibir un flujo de bits de datos codificados que representa una secuencia de vídeo, por ejemplo como resultado del método 800. En la etapa 901, un flujo de bits se recibe en un decodificador. El flujo de bits comprende un segmento codificado y una pluralidad de APS que incluye una pluralidad de tipos de APS. La pluralidad de tipos de APS puede incluir un tipo de ALF que contiene parámetros de ALF, un tipo de lista de escalado que contiene parámetros de lista de escalado, y un tipo de LMCS que contiene parámetros de LMCS. Por ejemplo, cada APS puede incluir un código de tipo de parámetro de APS (p. ej., `aps_params_type`) establecido a un valor predefinido que indica un tipo de parámetros incluidos en cada APS. Cada APS incluye un ID de APS. Además, los ID de APS para los tipos de APS se asignan en secuencia en una pluralidad de diferentes espacios de valores. Por consiguiente, cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS.

La pluralidad de diferentes espacios de valores puede superponerse. Por ejemplo, cada uno de los diferentes espacios de valores se puede extender en un intervalo predefinido. Los intervalos predefinidos se pueden determinar en función de los tipos de APS. Por ejemplo, un APS actual puede incluir un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual. El ID de APS actual se relaciona con un ID de APS anterior asociado con un APS anterior del mismo tipo que el APS actual. Sin embargo, el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual. Por tanto, cada APS se identifica por una combinación de un tipo de APS actual y un ID de APS actual.

El decodificador puede obtener los parámetros de la pluralidad de APS empleando los ID de APS y los tipos de APS. Por ejemplo, el decodificador puede emplear un encabezado relacionado con el segmento que referencia el APS de cada tipo. El decodificador puede entonces obtener los parámetros para el segmento en función de tales referencias. En la etapa 903, el segmento se decodifica usando parámetros de la pluralidad de APS. En la etapa 905, el decodificador puede reenviar el segmento para visualización como parte de una secuencia de vídeo decodificada.

La FIGURA 10 es un diagrama esquemático de un sistema de ejemplo 1000 para codificar una secuencia de vídeo de imágenes hasta un flujo de bits, tal como un flujo de bits 500, empleando una pluralidad de tipos de APS, tales como APS ALF 512, APS de lista de escalado 513, y/o APS de LMCS 514. El sistema 1000 puede ser implementa por un codificador y un decodificador, tal como un sistema de códec 200, un codificador 300, un decodificador 400 y/o un dispositivo de codificado de vídeo 700. Además, el sistema 1000 puede emplearse al implementar el método 100, 800 y/o 900, y el mecanismo 600.

El sistema 1000 incluye un codificador de vídeo 1002. El codificador de vídeo 1002 comprende un módulo de determinación 1001 para determinar una pluralidad de tipos de parámetros empleados para codificar un segmento. El codificador de vídeo 1002 comprende además un módulo de codificado 1003 para codificar el segmento en un flujo de bits. El módulo de codificado 1003 es además para codificar en el flujo de bits la pluralidad de tipos de parámetros en una pluralidad de APS al incluir la pluralidad de tipos de parámetros en una pluralidad de tipos de APS. El módulo de codificado 1003 es además para codificar ID de APS en la pluralidad de APS. El codificador de vídeo 1002 comprende además un módulo de asignación 1005 para asignar los ID de APS a cada uno del APS al asignar ID de APS para los tipos de APS en secuencia en una pluralidad de diferentes espacios de valores. El codificador de vídeo 1002 comprende además un módulo de almacenamiento 1006 para almacenar el flujo de bits para la comunicación hacia un decodificador. El codificador de vídeo 1002 comprende además módulo de transmisión 1007 para transmitir el flujo de bits que incluye la pluralidad de APS con la pluralidad de tipos de parámetros para soportar decodificar los segmentos en un decodificador. El codificador de vídeo 1002 puede configurarse además para realizar cualquiera de las etapas del método 800.

El sistema 1000 también incluye un decodificador de vídeo 1010. El decodificador de vídeo 1010 comprende un módulo de recepción 1011 para recibir un flujo de bits que comprende un segmento y una pluralidad de APS que incluye una pluralidad de tipos de APS, en donde cada APS incluye un ID de APS, y en donde ID de APS para los tipos de APS se asignan en secuencia sobre una pluralidad de diferentes espacios de valores. El decodificador de vídeo 1010 comprende además un módulo de decodificación 1013 para decodificar el segmento usando parámetros de la pluralidad de APS. El decodificador de vídeo 1010 comprende además un módulo de reenvío 1015 para reenviar el segmento para visualización como parte de una secuencia de vídeo decodificada. El decodificador de vídeo 1010 puede configurarse además para realizar cualquiera de las etapas del método 900.

Un primer componente se acopla directamente 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 se acopla indirectamente al segundo componente cuando hay componentes intermedios distintos de 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 intervalo que incluye $\pm 10\%$ del número siguiente, a menos que se indique lo contrario.

También debe entenderse que las etapas de los métodos ejemplares expuestos en esta memoria no se requiere necesariamente que se realicen en el orden descrito, y el orden de las etapas de dichos métodos debe entenderse como meramente ejemplar. Asimismo, se pueden incluir etapas adicionales en tales métodos, y se pueden omitir o combinar ciertas etapas, en métodos congruentes con diversas realizaciones de la presente invención.

REIVINDICACIONES

1. Un método (900) implementado en un decodificador, el método comprende:

5 recibir (901) un flujo de bits que comprende sintaxis de datos de vídeo, en donde los datos de vídeo comprenden una pluralidad de imágenes;
 10 particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos codificados, en donde la sintaxis comprende una pluralidad de conjuntos de parámetros de adaptación, APS, que incluyen una pluralidad de tipos de APS asociados con un segmento codificado de los segmentos codificados, en donde cada APS incluye un identificador de APS, ID, y en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS;
 15 decodificar (903) el segmento codificado usando parámetros de la pluralidad de APS obtenidos en función de los ID de APS;
 en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, una correlación de luma con tipo de escalado de croma, LMCS, que contiene parámetros de LMCS;
 20 en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalar valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y
 en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS;
 25 en donde cada APS incluye un tipo de parámetros de APS, *aps_params_type*, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS.

2. El método (900) de la reivindicación 1, en donde los espacios de valor separado se superponen.

30 3. El método (900) de cualquiera de las reivindicaciones 1-2, en donde la pluralidad de tipos de APS incluye un tipo de lista de escalado que contiene parámetros de lista de escalado, en donde los parámetros de lista de escalado se incluyen en una lista de escalado para un proceso de transformada

35 4. El método (900) de cualquiera de las reivindicaciones 1-3, en donde un APS actual incluye un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual, en donde el ID de APS actual se relaciona con un ID de APS anterior asociado con un APS anterior de un mismo tipo que el APS actual, y en donde el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual.

40 5. El método (900) de cualquiera de las reivindicaciones 1-4, en donde cada APS se identifica por una combinación de un tipo de APS actual y un ID de APS actual.

6. Un método (800) implementado en un codificador, el método comprende:

45 recibir datos de vídeo que comprenden una pluralidad de imágenes;
 particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos;
 determinar (803) una pluralidad de tipos de parámetros empleados para codificar un segmento de la pluralidad de segmentos;
 50 codificar (801) el segmento hasta un flujo de bits como segmento codificado;
 codificar (805) la pluralidad de tipos de parámetros en una pluralidad de conjuntos de parámetros de adaptación, APS, del flujo de bits al incluir la pluralidad de tipos de parámetros en una pluralidad de tipos de APS;
 asignar (807) un identificador de APS, ID, a cada uno de los APS, en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS;
 55 codificar (809) cada ID de APS en la pluralidad de APS;

60 en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, una correlación de luma con tipo de escalado de croma, LMCS, que contiene parámetros de LMCS;
 en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalar valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y
 en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS;

en donde cada APS incluye un tipo de parámetros de APS, `aps_params_type`, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS.

- 5 7. El método de la reivindicación 6, en donde los espacios de valor separado se superponen.
8. El método de cualquiera de las reivindicaciones 6-7, en donde la pluralidad de tipos de APS incluye un tipo de lista de escalado que contiene parámetros de lista de escalado.
- 10 9. El método (800) de cualquiera de las reivindicaciones 6-8, en donde un APS actual incluye un ID de APS actual seleccionado de un intervalo predefinido en un espacio de valores actual, en donde el ID de APS actual se relaciona con un ID de APS anterior asociado con un APS anterior de un mismo tipo que el APS actual, y en donde el ID de APS actual no se relaciona con otro ID de APS anterior asociado con otro APS anterior de un tipo diferente al APS actual.
- 15 10. El método (800) de cualquiera de las reivindicaciones 6-9, en donde cada APS se identifica por una combinación de un tipo de APS actual y un ID de APS actual.
- 20 11. Un medio no transitorio legible por ordenador que comprende un producto de programa informático para su uso por un dispositivo de codificado de vídeo, el producto de programa informático comprende instrucciones ejecutables por ordenador almacenadas en el medio no transitorio legible por ordenador de tal manera que cuando es ejecutado por un procesador hace que el dispositivo de codificado de vídeo realice el método de cualquiera de las reivindicaciones 1-10.
- 25 12. Un decodificador (1010) que comprende:
una unidad de recepción (1011), configurada para:
recibir un flujo de bits que comprende sintaxis de datos de vídeo, en donde los datos de vídeo comprenden una pluralidad de imágenes;
una unidad de partición, configurada para: particionar una imagen de la pluralidad de imágenes en una pluralidad de segmentos codificados, en donde la sintaxis comprende una pluralidad de conjuntos de parámetros de adaptación, APS, que incluyen una pluralidad de tipos de APS asociados con un segmento codificado de los segmentos codificados, en donde cada APS incluye un identificador de APS, ID, y en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS; y
unidad de decodificación (1013), configurada para decodificar el segmento codificado usando parámetros de la pluralidad de APS obtenidos en función de los ID de APS;
en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, una correlación de luma con tipo de escalado de croma, LMCS, que contiene parámetros de LMCS;
en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalar valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y
en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS;
en donde cada APS incluye un tipo de parámetros de APS, `aps_params_type`, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS.
- 30 13. El decodificador (1010) de la reivindicación 12, en donde el decodificador se configura además para realizar el método de cualquiera de las reivindicaciones 1-5.
- 55 14. Un codificador (1002) que comprende:
una unidad de recepción, configurada para recibir datos de vídeo que comprenden una pluralidad de imágenes;
una unidad de particionado, configurada para particionar una imagen en la pluralidad de imágenes en una pluralidad de segmentos;
una unidad de determinación (1001), configurada para determinar una pluralidad de tipos de parámetros empleados para codificar un segmento;
una unidad de codificación (1003), configurada para:
60 codificar el segmento hasta un flujo de bits como segmento codificado;

codificar la pluralidad de tipos de parámetros en una pluralidad de conjuntos de parámetros de adaptación, APS, del flujo de bits al incluir la pluralidad de tipos de parámetros en una pluralidad de tipos de APS; y
 codificar cada identificador de APS, ID, en la pluralidad de APS; y

5

una unidad de asignación (1005), configurada para asignar un ID de APS a cada uno de los APS, en donde cada uno de los tipos de APS usa un espacio de valores separado para el ID de APS;

10

en donde la pluralidad de tipos de APS incluye un tipo de filtro de bucle adaptativo, ALF, que contiene parámetros de ALF, y una correlación de luma con tipo de escalado de croma, LMCS, que contiene parámetros de LMCS;

en donde el APS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a segmentos, los parámetros de LMCS se usan para un LMCS que es un proceso que escalar valores de croma en función de correlaciones de luma, y en donde los parámetros de ALF se usan para un ALF que es un filtro que es controlado por parámetros incluidos en el APS de ALF; y

15

en donde cada uno de los espacios de valores separados se extiende en un intervalo predefinido, y en donde los intervalos predefinidos se determinan en función de los tipos de APS;

en donde cada APS incluye un tipo de parámetros de APS, *aps_params_type*, establecido a un valor predefinido que indica el tipo de parámetro de cada APS, y el tipo de parámetro indica los parámetros de ALF o los parámetros de LMCS

20

15. El codificador (1002) de la reivindicación 14, en donde el codificador se configura además para realizar el método de cualquiera de las reivindicaciones 6-10.

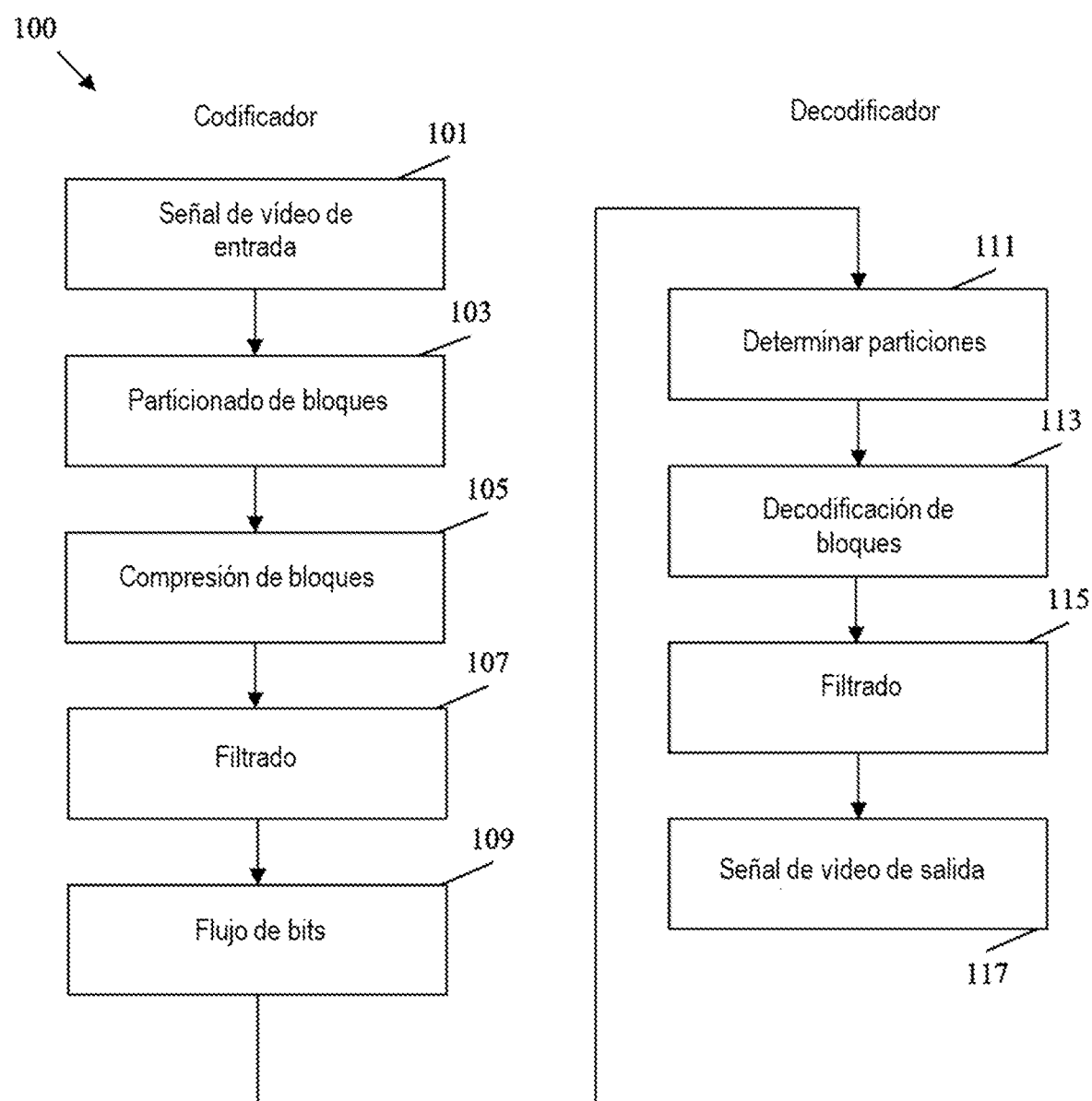


FIG. 1

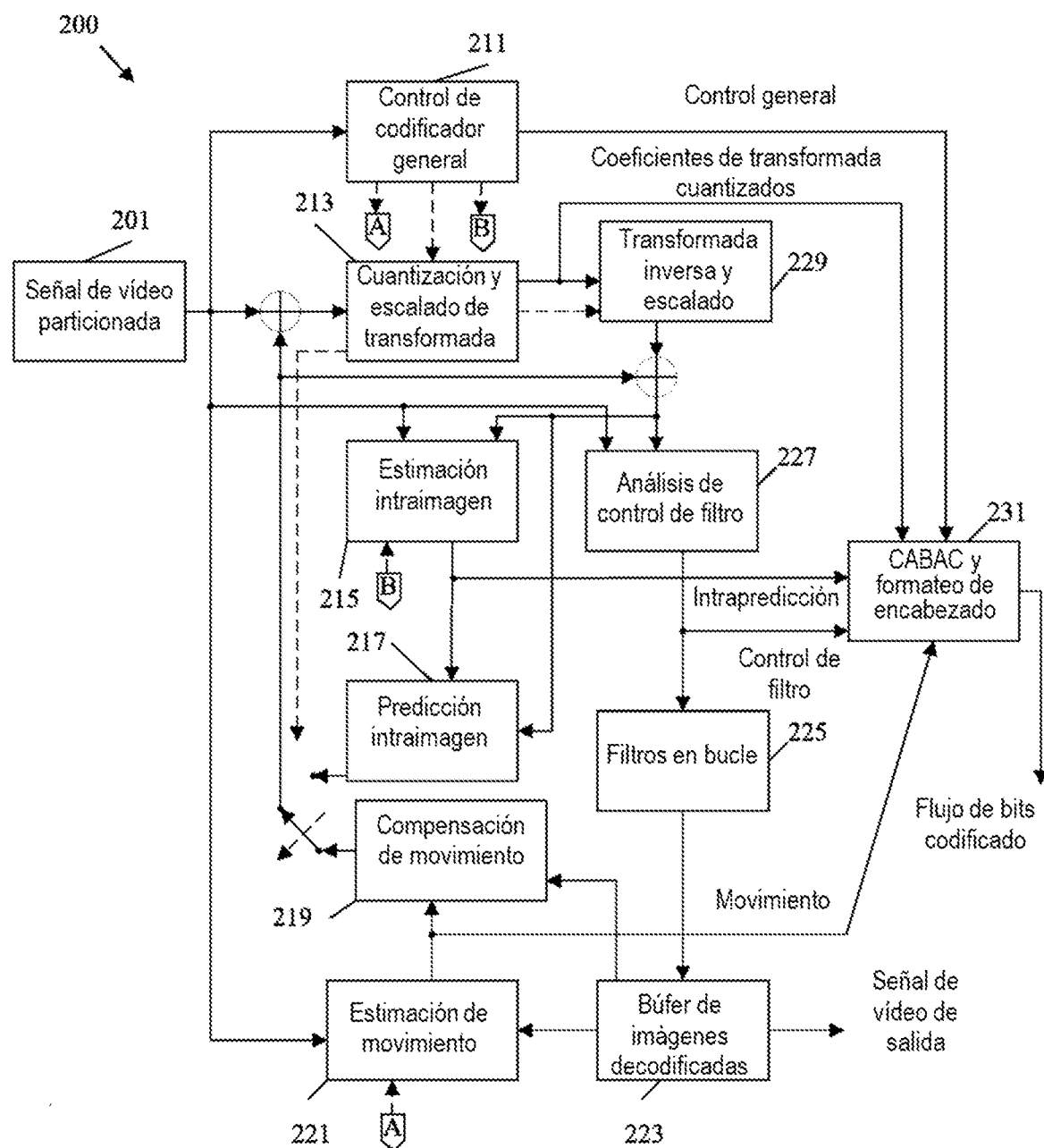


FIG. 2

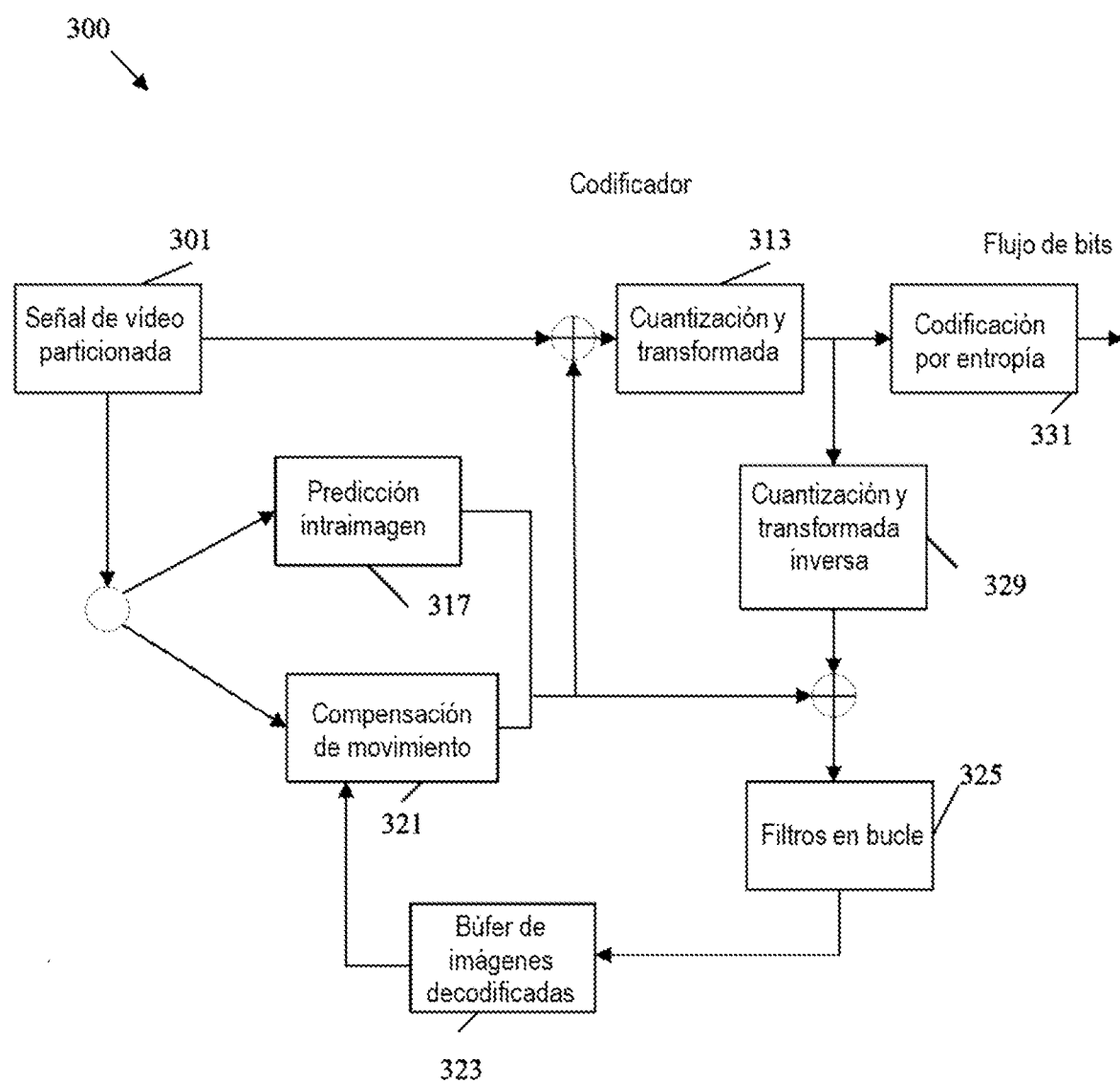


FIG. 3

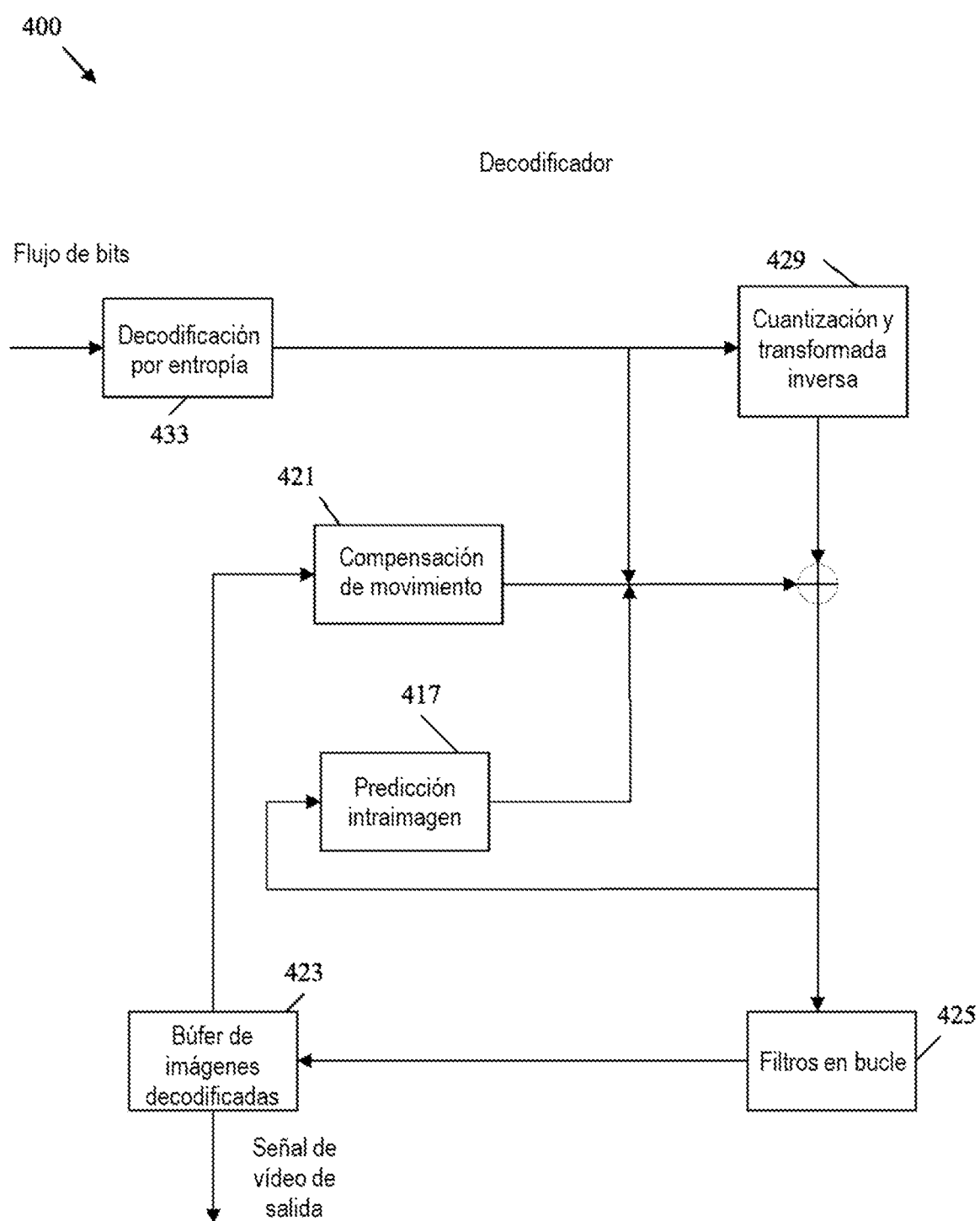


FIG. 4

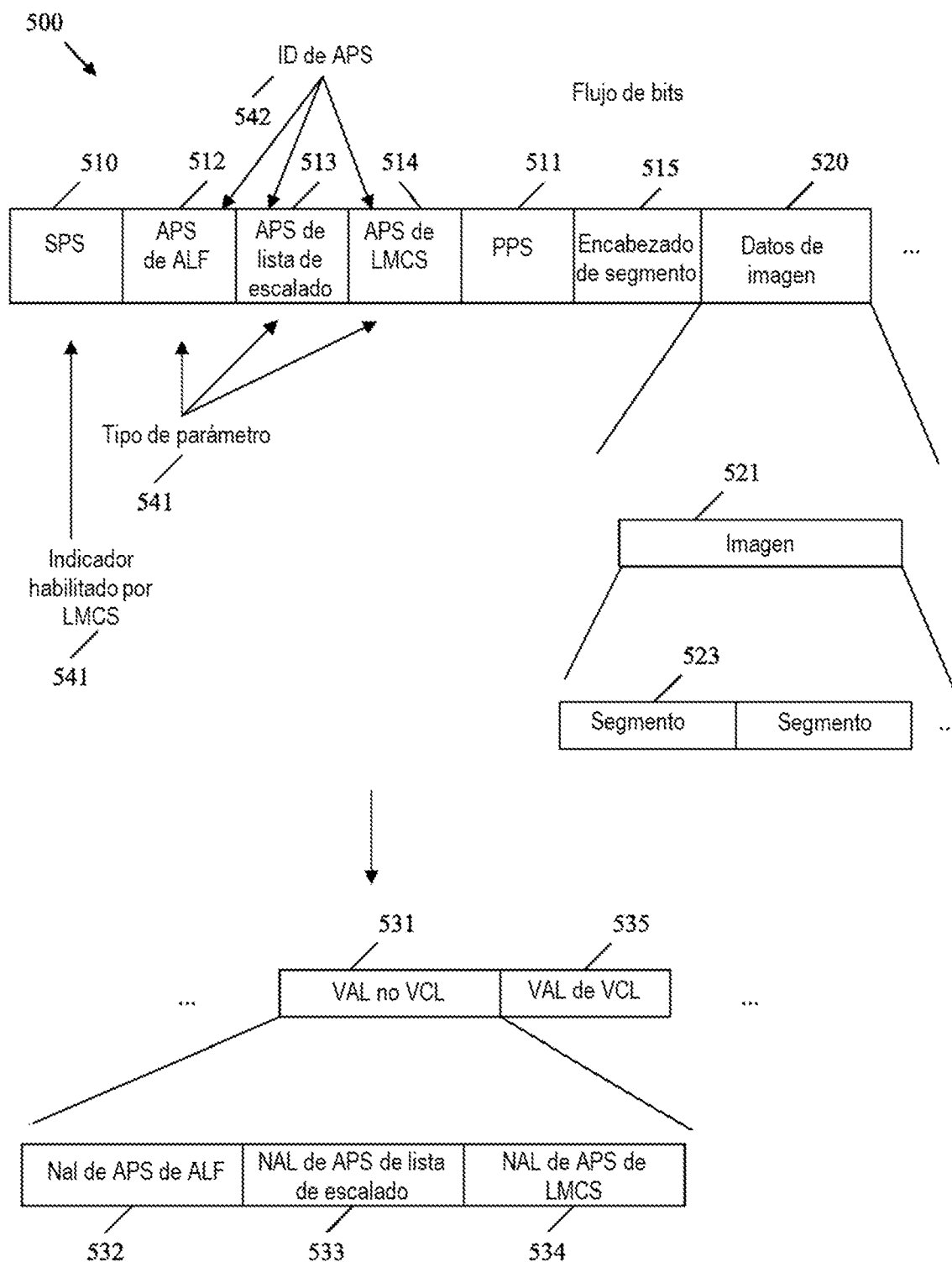


FIG. 5

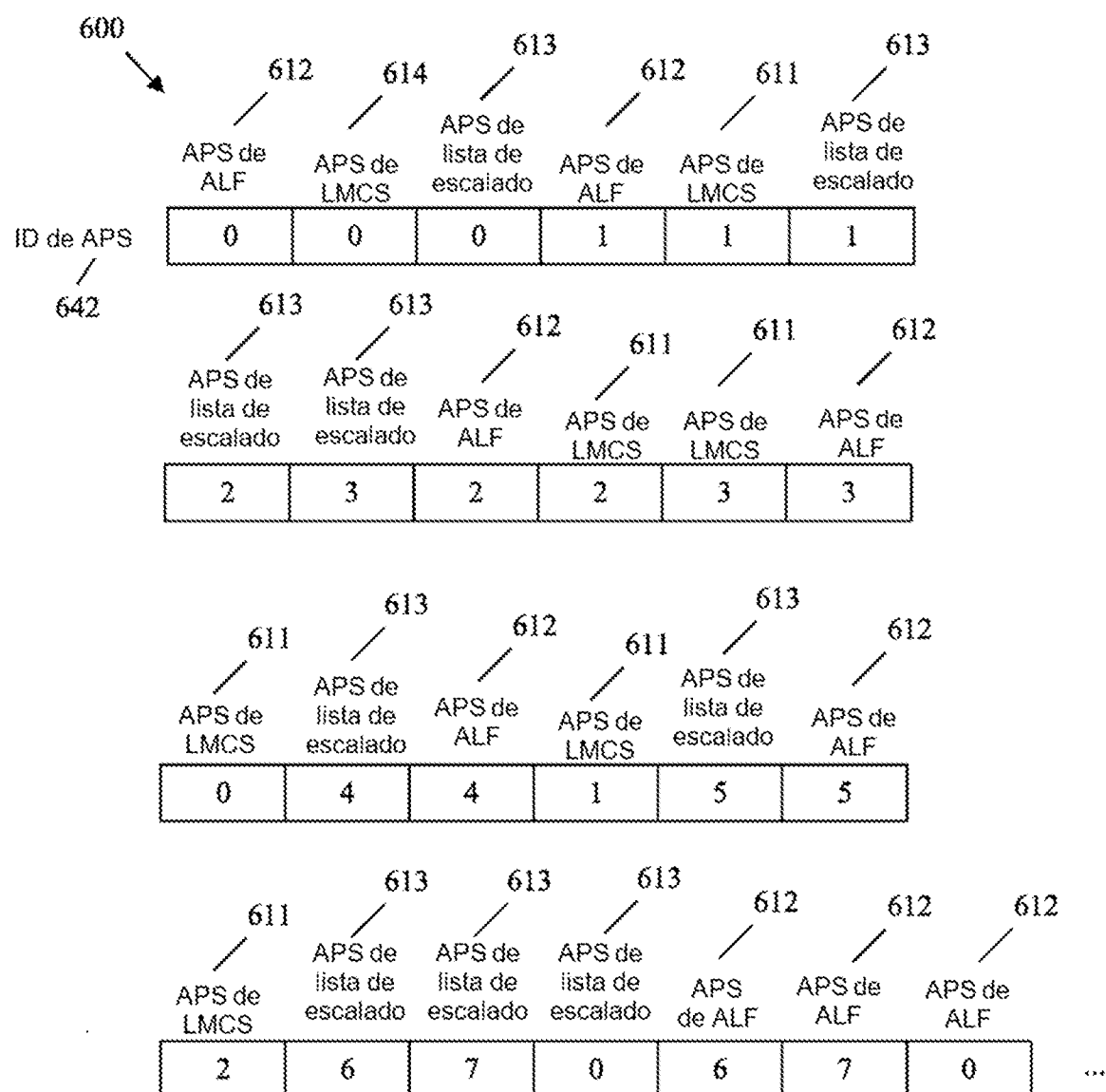


FIG. 6

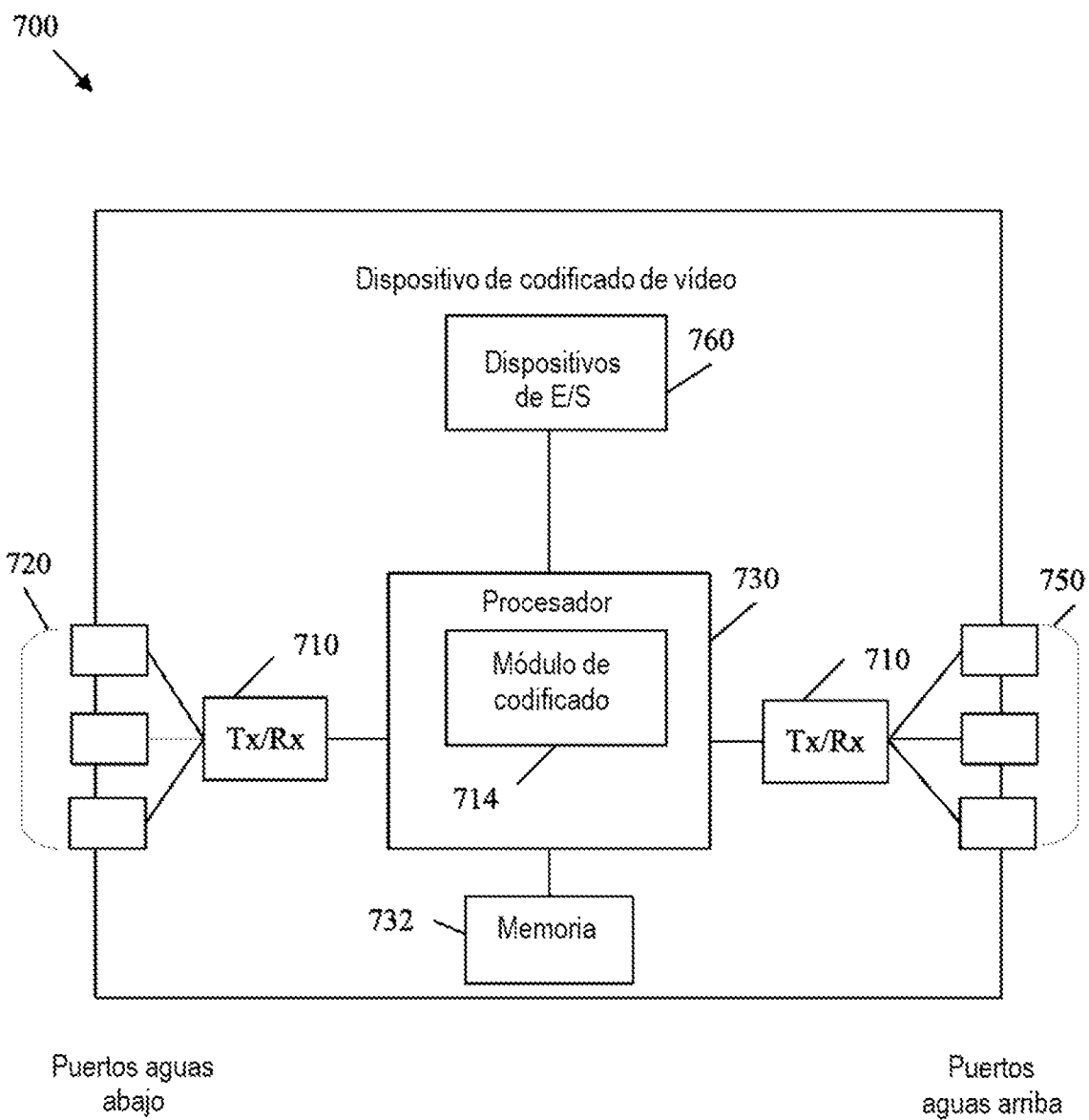


FIG. 7

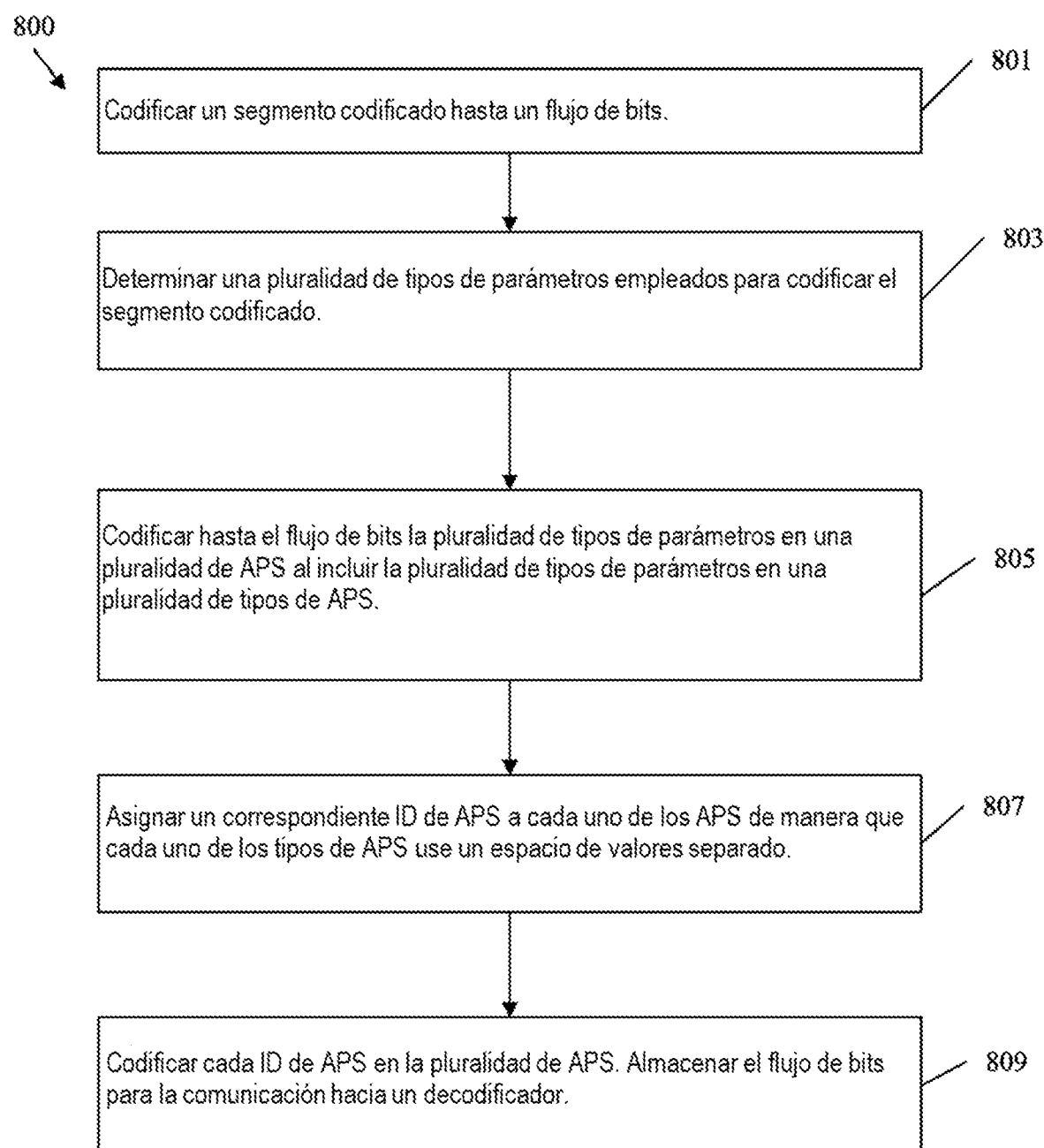


FIG. 8

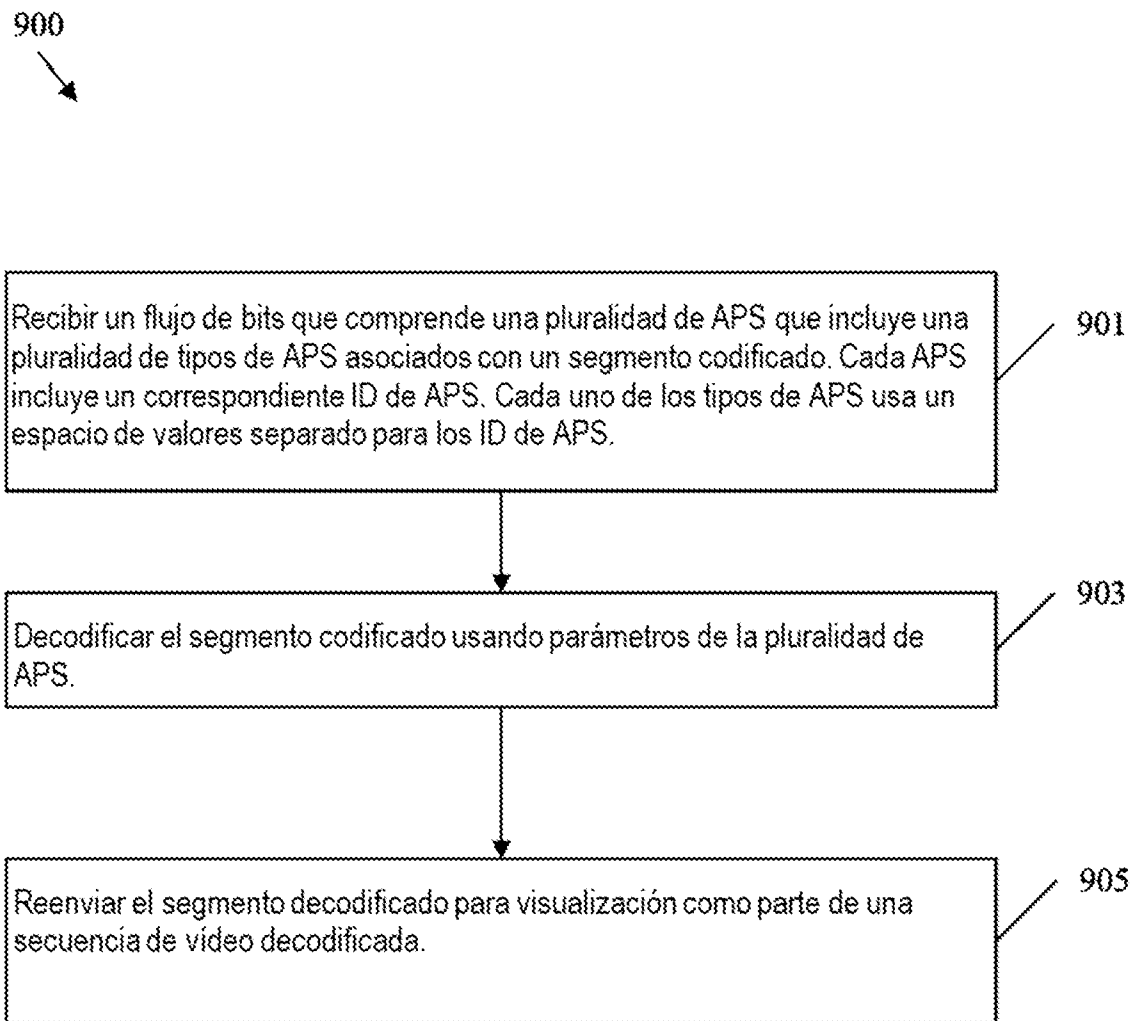


FIG. 9

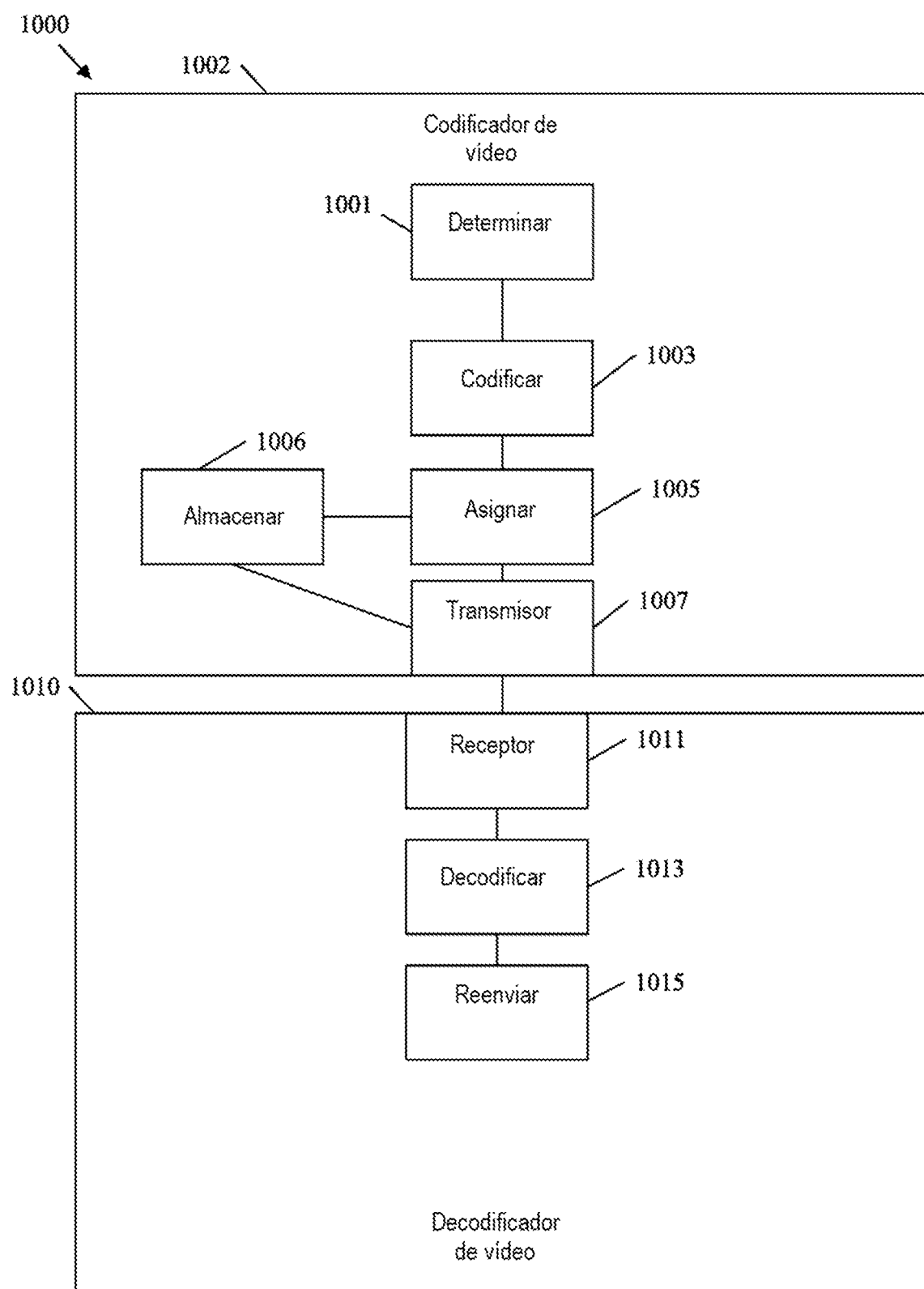


FIG. 10