

CODIFICACIÓN Y DECODIFICACIÓN DE VÍDEO QUE INVOLUCRAN

SUBIMÁGENES MAESTRAS

Campo técnico

Se divulgan realizaciones relacionadas con la codificación y
5 decodificación de vídeo.

Antecedentes

Codificación de vídeo versátil (VVC)

El estándar VVC y sus predecesores, Advanced Video Coding (AVC) y
High Efficiency Video Coding (HEVC), son códecs de video basados en bloques,
10 estandarizados y desarrollados conjuntamente por el ITU-T y el MPEG. Estos
códecs utilizan predicción tanto temporal como espacial para generar datos de
píxeles predichos. El VVC, el HEVC y el AVC son similares en muchos aspectos.
La predicción espacial se logra mediante predicción intra (I) desde el interior de
una imagen actual. La predicción temporal se logra mediante predicción inter
15 unidireccional (P) o bidireccional (B) a nivel de bloque, de imágenes de referencia
previamente decodificadas.

En el codificador, la diferencia entre los datos originales del píxel y los
datos de píxeles predichos, denominados residuos, se transforma en el dominio
de la frecuencia, se cuantifica y luego se codifica entropía (por ejemplo,
20 codificada usando codificación aritmética) antes de transmitirse junto con los
parámetros de predicción necesarios como el modo de predicción y los vectores
de movimiento, también codificados por entropía (por ejemplo, también

codificados aritméticamente). Para reconstruir una imagen, el decodificador obtiene el residuo de la imagen (es decir, realiza la decodificación (por ejemplo, decodificación aritmética), cuantificación inversa y transformación inversa para obtener el residuo) y luego suma el residuo de la imagen a una predicción intra o inter.

Vídeo, Imagen y Componentes

Un vídeo (también conocido como secuencia de vídeo) comprende una serie de imágenes en las que cada imagen consiste de uno o más componentes. Una imagen en una secuencia de vídeo a veces se denomina “imagen” o “trama”.

10 Cada componente de una imagen puede describirse como un arreglo rectangular bidimensional de valores de muestras de imagen (o “valores de muestras” para abreviar). Es común que una imagen en una secuencia de vídeo consista de tres componentes; un componente de luminancia Y donde los valores de muestra son valores de luminancia y dos componentes de croma Cb y Cr, donde los

15 valores de la muestra son valores de croma. Otras representaciones comunes incluyen ICtCb, IPT, YCbCr de luminancia constante, YCoCg y otras. También es común que las dimensiones de los componentes de croma sean menores que las de los componentes de luminancia por un factor de dos en cada dimensión. Por ejemplo, el tamaño del componente de luminancia de una imagen HD sería

20 1920x1080 y los componentes de croma tendrían cada uno la dimensión de 960x540. A veces se denomina a los componentes “componentes de color”, y otras veces “canales”.

Bloques y Unidades

En muchos estándares de codificación de vídeo, como HEVC y VVC, cada componente se divide en bloques y un flujo de bits de vídeo codificado producido por el codificador comprende una serie de bloques codificados. Un bloque es un arreglo bidimensional de valores de muestra (o “muestras” para abreviar). Es común en la codificación de vídeo que la imagen se divida en unidades que cubren un área específica de la imagen. Cada unidad consiste de todos los bloques de todos los componentes que conforman esa área específica y cada bloque pertenece completamente a una sola unidad. El macrobloque en H.264 y la unidad de codificación (CU) en HEVC son ejemplos de unidades.

Un bloque puede ser un “bloque de transformación” o un “bloque de predicción”. Un bloque de transformación es un bloque al que se le aplica una transformación utilizada en la codificación, y un “bloque de predicción” es un bloque al que se le aplica un único modo de predicción.

15 Estructura de bloque VVC

El estándar de codificación de video VVC utiliza una estructura de bloques denominada estructura de bloques de árbol cuaternario más árbol binario más árbol ternario (QTBT+TT), en la que cada imagen se particiona inicialmente en bloques cuadrados denominados unidades de árbol de codificación (CTU). El tamaño de todas las CTU es idéntico y la partición se realiza sin ningún elemento de sintaxis que la controle. Cada CTU se particiona adicionalmente en unidades de codificación (CU), que pueden tener forma cuadrada o rectangular. La CTU

se particiona primero mediante una estructura de árbol cuaternario, a continuación, puede particionarse adicionalmente en particiones de igual tamaño, ya sea vertical u horizontalmente, en una estructura binaria, para formar unidades de codificación (CU). Por lo tanto, un bloque podría tener forma

5 cuadrada o rectangular. La profundidad del árbol cuaternario y del árbol binario puede ser establecida por el codificador en el flujo de bits. La parte de árbol ternario (TT) añade la posibilidad de dividir una CU en tres particiones en lugar de dos particiones de igual tamaño; esto incrementa las posibilidades de utilizar una estructura de bloques que se ajuste mejor a la estructura del contenido de

10 una imagen.

Unidades de NAL

Tanto VVC como HEVC definen una Capa de Abstracción de Red (NAL). Todos los datos, es decir, tanto los datos de la Capa de Codificación de Vídeo (VCL) como los datos no VCL en HEVC y VVC, están encapsulados en unidades

15 NAL. Una unidad NAL VCL contiene datos que representan valores de muestra de imágenes. Una unidad NAL no VCL contiene datos asociados adicionales (también conocidos como metadatos) como conjuntos de parámetros y mensajes de información suplementaria de mejora (SEI). La unidad NAL en VVC y HEVC comienza con un encabezado llamado encabezado de unidad NAL. La sintaxis

20 del encabezado de la unidad NAL para HEVC comienza con un `forbidden_zero_bit`, que siempre debe ser igual a 0 para evitar la emulación de códigos de inicio. Sin este, algunos sistemas MPEG podrían confundir el flujo de

bits de video HEVC con otros datos, pero el bit 0 en el encabezado de la unidad NAL hace que todos los posibles flujos de bits HEVC sean identificables de manera única como flujos de bits HEVC. El encabezado de la unidad NAL en VVC es muy similar a la de HEVC, pero utiliza 1 bit menos para el `nal_unit_type` y, en su lugar, reserva este bit para uso futuro. Las palabras de código `nal_unit_type`, `nuh_layer_id` y `nuh_temporal_id_plus1` especifican el tipo de unidad NAL, de la unidad NAL que identifica qué tipo de datos se transporta en la unidad NAL, el identificador de capa de escalabilidad y el ID de capa temporal al que pertenece la unidad NAL. El tipo de unidad NAL indica y especifica cómo debe analizarse y decodificarse la unidad NAL. El resto de los bytes de la unidad NAL constituye la carga útil del tipo indicado por el tipo de unidad NAL. Un flujo de bits comprende una serie de unidades NAL concatenadas (es decir, un conjunto ordenado de unidades NAL).

Un decodificador puede concluir cómo debe manejarse la unidad NAL, por ejemplo, analizada y decodificada, tras observar el encabezado de la unidad NAL. Todos los flujos de bits VVC o HEVC consisten en una serie de unidades NAL concatenadas. El orden de decodificación es el orden en que deben decodificarse las unidades NAL, que es el mismo que el orden de las unidades NAL dentro del flujo de bits. El orden de decodificación puede ser diferente del orden de emisión, que es el orden en que las imágenes decodificadas deben ser generadas, como para la visualización, por el decodificador.

Segmentos

El concepto de segmentos en HEVC divide una imagen en segmentos codificados de forma independiente, donde la decodificación de un segmento en una imagen es independiente de otros segmentos de la misma imagen.

- 5 Se podían usar diferentes tipos de codificación para segmentos de la misma imagen, es decir, un segmento podía ser un segmento I, un segmento P, o un segmento B. Uno de los propósitos de los segmentos es permitir la resincronización en caso de pérdida de datos. En HEVC, un segmento es un conjunto de CTU.
- 10 En VVC, un segmento se define como un número entero de teselas completas o un número entero de filas CTU completas consecutivas dentro de una tesela de una imagen que están contenidas exclusivamente en una única unidad NAL. En VVC, una imagen puede particionarse en segmentos de barrido progresivo o en segmentos rectangulares. Un segmento de barrido progresivo
- 15 comprende un número de teselas completas en orden de barrido progresivo. Un segmento rectangular consiste de un grupo de teselas que juntas ocupan una región rectangular en la imagen o un número consecutivo de filas CTU dentro de una tesela. Cada segmento tiene un encabezado de segmento que comprende elementos de sintaxis. Se utilizan valores de encabezado de segmento
- 20 decodificados de estos elementos sintácticos al decodificar el segmento. Cada segmento se transporta en una unidad NAL VCL (también conocida como unidad

NAL de segmento codificada). En un borrador inicial de la especificación VVC, los segmentos se denominaban grupos de teselas.

Estructura de segmentos y macrobloques en estándares anteriores

Los estándares H.263 y H.264 utilizan la entidad macrobloque, que es una
5 unidad de procesamiento de bloque de 16x16 y definen un segmento como un número entero de macrobloques en un orden particular. H.263 y H.264 usan el término segmento en un contexto diferente en comparación con HEVC y VVC.

El Anexo K de la especificación H.263 describe el modo opcional de estructura en segmentos que incluye dos submodos:

10 (1) El submodo de Segmento Rectangular (RS). Cuando se utiliza RS, el segmento debe ocupar una región rectangular de ancho especificado por el parámetro de indicación de ancho de segmento (SWI) del encabezado del segmento en unidades de macrobloques, y contiene varios macrobloques en orden de barrido dentro de la región rectangular. Cuando el submodo de
15 Segmento Rectangular no está en uso, el campo SWI no está presente en el encabezado de segmento y un segmento contiene varios macrobloques en orden de barrido dentro de la imagen en su conjunto.

(2) El submodo de Orden de Segmento Arbitrario (ASO). Cuando se utiliza ASO, los segmentos pueden aparecer en cualquier orden dentro del flujo de bits.
20 Cuando ASO no está en uso, los segmento deben enviarse en el orden (único) en el que el campo de Dirección MacroBlock (MBA) del encabezado de

segmento aumenta estrictamente de cada segmento a cada segmento posterior en la imagen.

La especificación H.263 describe una indicación de macrobloque codificada (COD) como un bit al inicio de la sintaxis del macrobloque, que cuando se configura en “0” señala que el macrobloque está codificado y, si está en “1”, señala que no se transmite más información para este macrobloque y el macrobloque se maneja como “omitido”. Los macrobloques omitidos en H.263 se decodifican como macrobloques INTER con vector de movimiento para todo el bloque igual a cero y sin datos de coeficientes.

La especificación H.264 (T-REC-H.264-202108) también define un macrobloque omitido como un macrobloque para el que no se codifican datos salvo una indicación de que el macrobloque debe ser decodificado como “omitido”. Esta indicación puede ser común a varios macrobloques. Los elementos como componentes del vector de movimiento e índices de referencia se derivan para el macrobloque omitido de acuerdo con algunos procedimientos de derivación definidos.

La semántica de datos del segmento en H.264 incluye:

TABLA 1

mb_skip_run especifica el número de macrobloques consecutivos omitidos para los cuales, al decodificar un segmento P o SP, se debe inferir que mb_type es P_Skip y el tipo de macrobloque se denomina colectivamente tipo de macrobloque P, o para los cuales, al decodificar un segmento B, se

inferirá que `mb_type` es `B_Skip` y el tipo de macrobloque se denomina colectivamente tipo de macrobloque B. El valor de `mb_skip_run` estará en el rango de 0 a `PicSizeInMbs - CurrMbAddr`, inclusive.

`mb_skip_flag` igual a 1 especifica que para el macrobloque actual, al decodificar un segmento P o SP, `mb_type` debe inferirse como `P_Skip` y el tipo de macrobloque se denomina colectivamente tipo de macrobloque P, o para el cual, al decodificar un segmento B, se inferirá que `mb_type` es `B_Skip` y el tipo de macrobloque se denomina colectivamente tipo de macrobloque B. `mb_skip_flag` igual a 0 especifica que el macrobloque actual no se omite.

Teselas

El estándar de codificación de vídeo VVC incluye una herramienta llamada teselas que divide una imagen en regiones rectangulares espacialmente independientes. Las teselas en VVC son similares a las que se usan en HEVC. Usando teselas, una imagen en VVC puede particionarse en filas y columnas de CTU donde una tesela es la intersección de una fila y una columna.

La estructura de teselas se señala en el conjunto de parámetros de imagen (PPS) especificando los grosores de las filas y los anchos de las columnas. Las filas y columnas individuales pueden tener diferentes tamaños, pero la partición siempre abarca toda la imagen, de izquierda a derecha y de arriba a abajo respectivamente.

No hay dependencia de decodificación entre teselas de la misma imagen. Esto incluye la predicción intra, la selección de contexto para codificación y la predicción del vector de movimiento. Una excepción es que generalmente se permiten dependencias de filtrado en bucle entre teselas.

- 5 En el modo de segmento rectangular en VVC, una tesela puede dividirse adicionalmente en múltiples segmentos donde cada segmento comprende un número consecutivo de filas CTU dentro de una sola tesela. La FIG. 1 A muestra un ejemplo de partición de tesela VVC y la FIG. 1B muestra un ejemplo de partición de segmentos rectangulares usando la partición de teselas VVC.

10 Subimágenes

- Las subimágenes se soportan en VVC, donde una subimagen se define como una región rectangular de uno o más segmentos dentro de una imagen. Esto significa que una subimagen contiene uno o más segmentos que cubren colectivamente una región rectangular de una imagen. En la especificación VVC,
- 15 la ubicación y el tamaño de la subimagen se señalizan en el conjunto de parámetros de secuencia. Los límites de una región subimagen pueden tratarse como límites de imagen (excluyendo las operaciones de filtrado en bucle) condicionados a un indicador por subimagen `subpic_treated_as_pic_flag[ i ]` en el SPS. Asimismo, el filtrado en bucle a través de los límites de subimagen está
- 20 condicionado a un indicador por subimagen `loop_filter_across_subpic_enabled_flag[ i ]` en el SPS.

Esto se hace con una primera palabra código, `sps_num_subpics_minus1`, que especifica cuántas subimágenes hay. Luego hay un bucle que para cada subíndice de imagen *i* especifica la ubicación y tamaño de una subimagen. Este bucle asocia cada subimagen a su valor de índice de subimagen *i*. El SPS

5 también especifica para cada subimagen si sus límites deben tratarse como límites de imagen o no para los procedimientos de compensación de movimiento y/o filtrado en bucle. Los límites de una región de subimagen pueden tratarse como límites de imagen (excluyendo las operaciones de filtrado en bucle, condicionados a un indicador por subimagen `subpic_treated_as_pic_flag[i]` en el

10 SPS. Asimismo, el filtrado en bucle en los límites de la subimagen está condicionado a un indicador por subimagen `loop_filter_across_subpic_enabled_flag[i]` en el SPS.

En el encabezado de segmento VVC, hay un elemento sintáctico de ID de subimagen y un elemento de sintaxis de dirección de segmento. El ID de la

15 subimagen especifica el ID de la subimagen de la subimagen y la dirección del segmento especifica la dirección del segmento dentro de la subimagen. Todos los segmento que pertenecen a la misma subimagen deben tener el mismo valor ID de subimagen. Para soportar operaciones de extracción y fusión de flujo de bits sin necesidad de reescribir encabezados de segmento, existe un mecanismo

20 de mapeo en VVC entre los valores ID de subimagen y los valores de índice de subimagen. Este mapeo se señala mediante los elementos sintácticos del `sps_subpic_id[i]` o del `pps_subpic_id[i]` en el SPS y el PPS, respectivamente. Por

ejemplo, un valor `sps_subpic_id[1]` igual a 3 significa que el valor del ID de subimagen 3 se mapea al valor del índice de subimagen 1. Mediante este mecanismo de mapeo, solo es necesario reescribir los elementos de la sintaxis SPS o PPS durante una operación de extracción o fusión de flujo de bits. La

5 Tabla 2 muestra la sintaxis de subimagen en el SPS en VVC.

TABLA 2 - Sintaxis de subimagen en el SPS en la especificación VVC

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>sps_subpics_info_present_flag</code>	<code>u(1)</code>
<code>if(sps_subpics_present_flag) {</code>	
<code>...</code>	
<code>sps_num_subpics_minus1</code>	<code>UE(v)</code>
<code>for(i = 0; sps_num_subpics_minus1 > 0 & i <=</code> <code>sps_num_subpics_minus1; i++) {</code>	
<code>...</code>	
<code>sps_subpic_ctu_top_left_x[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_ctu_top_left_y[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_width_minus1[i]</code>	<code>u(v)</code>
<code>...</code>	

sps_subpic_height_minus1[l]	u(v)
...	
sps_subpic_treated_as_pic_flag[l]	u(1)
sps_loop_filter_across_subpic_enabled_flag[l]	u(1)
...	
}	
}	
...	

Operaciones de extracción y fusión de flujo de bits

Algunas aplicaciones de vídeo implican operaciones realizadas sobre flujos de bits, incluyendo extracción de flujo de bits y operaciones de fusión de

5 flujo de bits. Una operación de extracción de flujo de bits toma un flujo de bits de entrada como entrada y produce un flujo de bits de emisión, donde el flujo de bits de emisión representa un subconjunto del contenido de vídeo del flujo de bits de entrada. Una operación de fusión de flujo de bits toma múltiples flujos de entrada como entrada y produce un único flujo de bits de emisión, donde el flujo de bits

10 de emisión representa la unión del contenido de vídeo de los flujos de datos de entrada. Para que la fusión de flujos de bits funcione, es muy común que los flujos de bits de entrada se creen pensando en el soporte de fusión. Las operaciones de fusión de flujo de bits en las que los flujos de bits se codifican

completamente de forma independiente, por ejemplo, por codificadores diferentes, normalmente no son posibles.

Es deseable que el formato de flujo de bits soporte operaciones de extracción y fusión de flujo de bits sin requerir cambios de flujo de bits de bajo nivel, ya que los cambios de flujo de bits de bajo nivel implican una carga de implementación considerable y una mayor complejidad computacional. Un objetivo común de disposición es habilitar estas operaciones sin necesidad de manipular datos codificados de segmentos. Por ejemplo, en VVC sería aceptable reescribir datos de conjuntos de parámetros, pero no aceptable reescribir datos de segmentos o datos de encabezados de imagen.

Para que la extracción y la fusión funcionen, el formato de vídeo debe soportar la partición espacial del contenido en segmentos codificados de forma independiente. Segmentos, teselas y subimágenes son ejemplos de estos segmentos. En HEVC, normalmente se utilizan conjuntos de teselas con restricción de movimiento (MCTS), donde los segmentos que contienen una o más teselas completas se codifican de modo que los segmentos sean independientes. Se presta especial atención a los procedimientos de compensación de movimiento HEVC, donde el codificador HEVC debe seleccionar cuidadosamente los parámetros de movimiento para que los segmentos sean independientes, de ahí el nombre MCTS. En VVC, se introdujo el tipo de partición subimagen, para lo cual el soporte para la extracción de flujo de bits y la gestión de fusiones menos complejos fue una razón clave. En

comparación con los MCTS HEVC, las subimágenes VVC son más eficientes en compresión y no requieren un cuidado especial relacionado con la compensación del movimiento.

Las operaciones de extracción y fusión de flujo de bits para subimágenes en VVC podrían, por ejemplo, incluir extraer una o más subimágenes de un primer flujo de bits, extraer una o más subimágenes de un segundo flujo de bits y fusionar las subimágenes extraídas en un nuevo tercer flujo de bits.

Recuento de orden de imagen (POC)

Las imágenes en HEVC se identifican por sus valores de recuento de orden de imagen (POC), también conocidos como valores POC completos. Cada segmento contiene una palabra código, `pic_order_cnt_lsb`, que será igual para todos los segmento en una imagen. `pic_order_cnt_lsb` también se conoce como los bits menos significativos (lsb) de la POC completa, ya que es una palabra código de longitud fija y solo se señalizan los bits menos significativos de la POC completa. Tanto el codificador como el decodificador llevan un control de las POC y asignan valores de POC a cada imagen codificada/decodificada. El `pic_order_cnt_lsb` puede ser señalado por 4-16 bits. Existe una variable `MaxPicOrderCntLsb` usada en HEVC que se establece al valor máximo de `pic_order_cnt_lsb` más 1. Esto significa que si se usan 8 bits para señalar el `pic_order_cnt_lsb`, el valor máximo es 255 y `MaxPicOrderCntLsb` se establece en $2^8 = 256$. El valor de recuento de orden de imagen de una imagen se llama

PicOrderCntVal en HEVC. Normalmente, PicOrderCntVal para la imagen actual se llama simplemente PicOrderCntVal.

Gestión de imágenes de referencia

La gestión de imágenes de referencia en HEVC se realiza utilizando
5 conjuntos de imágenes de referencia (RPS). El conjunto de imágenes de referencia es un conjunto de imágenes de referencia que se señala en los encabezados de segmento. Cuando el decodificador ha decodificado una imagen, esta se junta con su valor POC en una memoria intermedia de imagen decodificado (DPB). Al decodificar una imagen de subsecuencia, el decodificador
10 analiza la sintaxis RPS del encabezado de segmento y construye listas de valores POC de imagen de referencia.

La gestión de imagen de referencia en la especificación VVC difiere ligeramente de la de HEVC. En HEVC, el RPS se señala y las listas de imágenes de referencia para la predicción Inter se derivan del RPS. En la
15 especificación VVC, las listas de imágenes de referencia (RPL) se señalizan y se deriva el RPS. Sin embargo, en ambas especificaciones, se hace la señalización de qué imágenes conservar en el DPB, cuáles deben ser a corto plazo y largo plazo. El uso de POC para la identificación de imágenes y la determinación de imágenes de referencia faltantes se hace igual en ambas
20 especificaciones.

Conjuntos de parámetros

HEVC y VVC especifican tres tipos de conjuntos de parámetros: el conjunto de parámetros de imagen (PPS), el conjunto de parámetros de secuencia (SPS) y el conjunto de parámetros de vídeo (VPS). El PPS contiene
5 datos comunes para una imagen completa, el SPS contiene datos comunes para una secuencia de vídeo codificado (CVS) y el VPS contiene datos comunes para múltiples CVS, por ejemplo, datos para múltiples capas de escalabilidad en el flujo de bits.

VVC también especifica un conjunto de parámetros adicional, el conjunto
10 de parámetros de adaptación (APS). El APS lleva los parámetros necesarios para la herramienta de filtro de bucle adaptativo (ALF), la herramienta de mapeo de luminancia y escalado de croma (LMCS) y la herramienta de lista de escalado.

Tanto HEVC como VVC permiten que cierta información (por ejemplo, conjuntos de parámetros) se proporcione por medios externos. “Por medios
15 externos” debe interpretarse como que la información no se proporciona en el flujo de bits de vídeo codificado, sino por algún otro medio no especificado en la especificación del códec de vídeo, por ejemplo, mediante metadatos posiblemente proporcionados en un canal de datos diferente, como una constante en el decodificador, o proporcionada mediante una API al
20 decodificador.

Información de capacidad de decodificación (DCI)

En VVC hay una unidad DCI NAL. El DCI especifica información que no cambia durante la sesión de decodificación y puede ser buena para que el decodificador la conozca desde el principio como de antemano, como la información de perfil y nivel. La información en el DCI no es necesaria para el funcionamiento del procedimiento de decodificación. En los borradores de la especificación VVC, el DCI se denominaba conjunto de parámetros de decodificación (DPS).

La información de capacidad de decodificación también puede contener un conjunto de restricciones generales para el flujo de bits, que proporcionan al decodificador información sobre qué esperar del flujo de bits, en términos de herramientas de codificación, tipos de unidades NAL, etc. En VVC, la información general de restricciones puede señalizarse en el DCI, VPS o SPS.

Encabezado de imagen

En VVC, una imagen codificada contiene una estructura de encabezado de imagen. La estructura del encabezado de imagen contiene elementos sintácticos comunes a todos los segmentos de la imagen asociada. La estructura del encabezado de imagen puede señalizarse en su propia unidad NAL no VCL con unidad NAL tipo PH_NUT o incluirse en el encabezado de segmento dado que solo hay un segmento en la imagen codificada. Esto se indica mediante el elemento de sintaxis de encabezado de segmento, `picture_header_in_slice_header_flag`, donde un valor igual a 1 especifica que la

estructura del encabezado de imagen está incluida en el encabezado de segmento y un valor igual a 0 especifica que la estructura del encabezado de imagen está transportada en su propia unidad PH NAL. Para un CVS donde no todas las imágenes son imágenes de un solo segmento, cada imagen codificada debe ir precedida por un encabezado de imagen que se señala en su propia unidad NAL. HEVC no soporta encabezados de imagen.

Capas, OLS y OPI

El valor del elemento sintáctico `nuh_layer_id` en el encabezado de la unidad NAL en VVC y HEVC especifica el ID de capa al que pertenece la unidad NAL. En VVC, las capas pueden codificarse de forma independiente o dependiente unas de otras. Cuando las capas se codifican de forma independiente, una capa con, por ejemplo, `nuh_layer_id` 0 puede no predecir datos de vídeo de otra capa con, por ejemplo, `nuh_layer_id` 1. En VVC, puede usarse codificación dependiente entre capas, lo que permite soporte para codificación escalable con SNR, escalabilidad espacial y de vista.

El VPS en VVC y HEVC contiene información de capa de escalabilidad necesaria para manejar flujos de bits escalables. Para HEVC, debe haber un VPS para todos los flujos de bits, incluso los de una sola capa, mientras que para VVC el VPS puede omitirse para flujos de bits de una sola capa. El VPS define conjuntos de capas de emisión (OLS), donde un OLS es una descripción de un conjunto de capas en el flujo de bits con indicaciones de cuál de las capas en el OLS debe ser de emisión. En VVC, solo se especifican las capas de emisión, y

el OLS completo se deriva utilizando información de flujo de bits para cómo las capas referencian otras capas. Esto significa que el OLS completo es decodificable incluso si se descartan todas las capas que no están incluidas en el OLS. En otras palabras, ninguna capa en un OLS depende de ninguna capa que no esté en el OLS. Esto también significa que puede ser necesario decodificar algunas capas OLS, pero no se emiten imágenes de esas capas.

La información del punto de operación (OPI) en VVC puede usarse para especificar el índice OLS del OLS objetivo para que el decodificador lo decodifique. La OPI puede además especificar la subcapa temporal más alta que el decodificador debe decodificar. Cuando un flujo de bits contiene muchas capas y subcapas, la OPI puede ser útil para indicar al decodificador qué partes del flujo de bits debe decodificar y/o qué subcapas temporales deben descartarse al decodificar. El OLS objetivo y la subcapa temporal más alta a decodificar pueden ser especificados alternativamente por medios externos. Si eso ocurre y hay un OLS en el flujo de bits, el decodificador debería usar la información proporcionada por medios externos e ignorar la información OLS. La OPI se señala en su propia unidad NAL no VCL en VVC.

Sumario

Actualmente existen ciertos desafíos. Por ejemplo, un problema con la tecnología existente descrita anteriormente es que todas las porciones de una imagen tienen el mismo “estado”, lo que significa que en el procedimiento de decodificación todos los valores de muestra decodificados en la imagen están

igualmente “valorados”. La tecnología existente no permite decodificar solo una porción de la imagen y abstenerse de decodificar otra porción de la imagen para un decodificador conforme. En otras palabras, la noción de que diferentes porciones de la imagen tengan distinta prioridad, importancia o estados no se reconoce en la tecnología existente. Por ello, la tecnología existente no permite que un decodificador conforme decodifica solo porciones de la imagen basándose, por ejemplo, en la cantidad de recursos (por ejemplo, memoria o potencia de procesamiento) disponibles para el decodificador, la configuración del dispositivo y/o preferencias.

Otro problema con la tecnología existente es que los mecanismos de omisión actúan únicamente como herramientas de codificación. Por ejemplo, el concepto de macrobloques omitidos en H.263 y H.264 se basa en abstenerse de señalar valores de muestra para un macrobloque pero codificar una indicación de omisión y, entonces en el lado del decodificador, tener un procedimiento de decodificación para decodificar el macrobloque como “omitido”, incluyendo un procedimiento de derivación para componentes del vector de movimiento e índices de referencia para los macrobloques omitidos con el fin de decodificar los valores de muestra. Este uso de macrobloques omitidos como herramienta de codificación solo ahorra bits cuando el macrobloque puede reconstruirse en el lado del decodificador mediante medios distintos a la señalización de los valores de la muestra, en lugar de proporcionar flexibilidad al decodificador para decodificar solo porciones de la imagen indicadas con un “estado” particular en

el flujo de bits. También se puede decir que la indicación de omisión en H.263 y H.264 representa realmente una opción o modo de codificación para el macrobloque. En ese sentido, el macrobloque se codifica usando el modo más barato en términos de coste de bits. No es el caso que los macrobloques omitidos
5 no estén codificados, sino que se codifican mediante la indicación de omisión.

Otro problema de la tecnología existente es que los mecanismos de omisión aplicables a porciones de la imagen son mecanismos de ahorro de bits y no un mecanismo para facilitar casos de uso con bajo retardo. No está soportado en los estándares existentes de codificación de imagen y vídeo no
10 decodificar porciones de la imagen para satisfacer los requisitos de bajo retardo o baja latencia.

En aplicaciones como videoconferencias, retransmisión deportiva y observación remota de una misión crítica, existen escenarios de uso en los que porción de la imagen es la parte más importante del contenido y, desde la
15 perspectiva del usuario, es preferible poder tener esa porción de la imagen decodificada y mostrada continuamente para cada imagen a no decodificar y mostrar fotogramas completos. En la tecnología existente, en situaciones como cuando se pierden una o más unidades NAL VCL en una imagen, o los recursos del decodificador se vuelven escasos en algún momento, o en caso de
20 problemas de conectividad o ancho de banda durante una sesión, una o más imágenes completas podrían no decodificarse y, en consecuencia, no mostrarse

completamente, lo que provoca pausas y saltos en la secuencia de vídeo decodificada y mostrada.

El contenido de vídeo en algunas aplicaciones tiene varias porciones independientes en un mismo fotograma. La tecnología existente, como MCTS en HEVC o segmentos y subimágenes independientes en VVC, facilita una
5 codificación y decodificación más eficiente y flexible de este tipo de contenido, pero aun así una imagen debe decodificarse completamente en todas esas porciones independientes antes de mostrarse, por lo que no es posible decodificar y mostrar solo partes selectivas de la imagen.

10 En consecuencia, en un aspecto se proporciona un procedimiento para decodificar una imagen de un flujo de bits, la imagen que comprende una primera porción y una segunda porción. El procedimiento incluye obtener, del flujo de bits, la primera información de estado (por ejemplo, un valor del primer indicador) correspondiente a la primera porción de la imagen, en la que la primera
15 información de estado no corresponde a la segunda porción. El procedimiento también incluye determinar si decodificar o no la primera porción de la imagen basándose en la primera información de estado.

En otro aspecto, se proporciona un procedimiento para producir un flujo de bits que comprende una imagen (es decir, una imagen codificada), la imagen
20 que comprende una primera porción y una segunda porción. El procedimiento incluye determinar si la primera porción debe ser decodificada por un decodificador. El procedimiento también incluye, como resultado de determinar

que la primera porción debe ser decodificada por un decodificador, produciendo información de estado codificada para la primera porción e incluir la información de estado codificada en el flujo de bits, donde la información de estado codificada para la primera porción no corresponde a la segunda porción, y la información de estado codificada sirve para informar al decodificador de que este no debe abstenerse de decodificar la primera porción de la imagen.

En otro aspecto, se proporciona un procedimiento realizado por un nodo de red. El procedimiento incluye recibir i) una primera unidad (por ejemplo, una primera unidad NAL VCL) que comprende valores de muestra de imagen codificada correspondientes a una primera porción de una imagen y ii) primera información de estado (por ejemplo, primera información de estado codificada) correspondiente a la primera porción de la imagen, en la que la primera información de estado no corresponde a una segunda porción de la imagen. El procedimiento también incluye recibir una segunda unidad (por ejemplo, una segunda unidad NAL VCL) que comprende valores de muestra de imagen codificada correspondientes a la segunda porción de la imagen. El procedimiento también incluye, basándose en la primera información de estado, decidir si se reenvía o no la primera unidad a un dispositivo receptor.

En otro aspecto, se proporciona un programa informático que comprende instrucciones que, al ejecutarse mediante la circuitería de procesamiento de un aparato, hace que el aparato realice cualquiera de los procedimientos aquí revelados. En una realización, se proporciona una portadora que contiene el

programa informático, en la que la portadora es una de una señal electrónica, una señal óptica, una señal de radio y un medio de almacenamiento legible por ordenador. En otro aspecto, se dispone de un aparato que está configurado para realizar los procedimientos aquí revelados. El aparato puede incluir memoria y
5 circuitería de procesamiento acoplados a la memoria.

Una ventaja de las realizaciones aquí divulgadas es que ofrecen flexibilidad para elegir qué porciones de una imagen decodificar, lo cual es útil cuando no todas las porciones de la imagen son de interés o cuando no hay recursos suficientes para decodificar la imagen completa. Esto introduce un
10 nuevo compromiso diferente a la tecnología existente, en la que solo se pueden decodificar imágenes completas.

Las realizaciones además tienen la ventaja de asignar recursos (tiempo, memoria) a las porciones más críticas de la imagen, por ejemplo en una configuración de bajo retardo y baja latencia. Si el decodificador no va a
15 decodificar al menos algunas porciones de una imagen, esa porción de la señal puede omitirse por completo, por lo que se podrían usar memoria intermedia más pequeñas.

Otra ventaja es que, si solo se van a mostrar o usar algunas porciones de la imagen en una aplicación y no la imagen completamente decodificada,
20 entonces el decodificador solo puede decodificar las porciones necesarias de la imagen y no las innecesarias. La solución propuesta en este caso será más

eficiente en recursos y reducirá el procesamiento y el consumo energético innecesarios.

Otra ventaja es que la experiencia del usuario mejora en ciertas aplicaciones al priorizar la decodificación y visualización de la porción crítica de la imagen (por ejemplo, en un vídeo deportivo la porción crítica puede ser el jugador con la pelota) en lugar de arriesgarse a no poder decodificar fotogramas completos debido a causas como pérdida de paquetes o reducción repentina de ancho de banda, especialmente en una configuración de bajo retardo.

Breve descripción de los dibujos

Los dibujos adjuntos, que aquí se incorporan y forman parte de la especificación, ilustran diversas realizaciones.

La FIG. 1A muestra un ejemplo de partición de una tesela en VVC.

La FIG. 1B muestra un ejemplo de partición de segmentos rectangulares usando la partición de tesela en VVC.

LA FIG. 1C ilustra un sistema de acuerdo con una realización.

La FIG. 2 es un diagrama de bloques esquemático de un codificador de acuerdo con una realización.

La FIG. 3 es un diagrama esquemático de bloques de un decodificador de acuerdo con una realización.

La FIG. 4A ilustra una imagen de ejemplo.

La FIG. 4B ilustra un ejemplo de un primer conjunto de porciones y un segundo conjunto de porciones.

La FIG. 5 ilustra un ejemplo donde dos disposiciones diferentes de subimagen usando la misma disposición de subimagen maestra indican dos porciones distintas de la imagen con estados maestro.

La FIG. 6 ilustra un ejemplo de cómo una actualización en la disposición
5 de la subimagen maestra afecta a la porción de la imagen con estado maestro.

La FIG. 7 ilustra un ejemplo donde la imagen se particiona en subimágenes 0 a 5 y las subimágenes 0 a 2 se indican como subimágenes maestras.

La FIG. 8 ilustra ejemplos de áreas maestras.

10 La FIG. 9 ilustra que una subimagen se indica como subimagen maestra si se superpone total o parcialmente con la parte de la imagen indicada como área maestra.

La FIG. 10 ilustra una imagen de textura y profundidad con un ROI.

La FIG. 11 ilustra un sistema de acuerdo con una realización.

15 La FIG. 12 es un diagrama de flujo que ilustra un procedimiento de acuerdo con una realización.

La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de acuerdo con una realización.

La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de
20 acuerdo con una realización.

La FIG. 15 es un diagrama esquemático de bloques de un decodificador de acuerdo con una realización.

La FIG. 16 es un diagrama de bloques de un aparato de codificación de acuerdo con una realización.

Descripción detallada

La FIG. 1C ilustra un sistema 100 de acuerdo con una realización. El sistema 100 incluye un codificador 102 y un decodificador 104, en los que el codificador 102 está en comunicación con el decodificador 104 a través de una red 110 (por ejemplo, Internet u otra red). El codificador 102 codifica una secuencia de vídeo fuente 101 en un flujo de bits que comprende una secuencia de vídeo codificada y transmite el flujo de bits al decodificador 104 a través de la red 110. En algunas realizaciones, el codificador 102 no está en comunicación con el decodificador 104 y, en tal realización, en lugar de transmitir el flujo de bits al decodificador 104, el flujo de bits se almacena en una unidad de almacenamiento de datos. El decodificador 104 decodifica las imágenes incluidas en la secuencia de vídeo codificada para emitir datos de vídeo para visualización y/o procesamiento posterior de imágenes (por ejemplo, una tarea de visión artificial). En consecuencia, el decodificador 104 puede formar parte de un dispositivo 103 que tenga un procesador de imagen 105 y/o una pantalla 106. El procesador de imagen 105 puede realizar tareas de visión artificial sobre las imágenes decodificadas. Una de esas tareas de visión artificial puede ser identificar objetos en la imagen. El dispositivo 103 puede ser un dispositivo móvil, un decodificador, una pantalla montada en la cabeza o cualquier otro dispositivo.

La FIG. 2 ilustra componentes funcionales del codificador 102 de acuerdo con algunas realizaciones. Cabe señalar que los codificadores pueden implementarse de forma diferente, por lo que se puede usar una implementación distinta a este ejemplo específico. El codificador 102 emplea un restador 241
5 para emitir un bloque residual, que es la diferencia en los valores de muestra entre un bloque de entrada y un bloque de predicción (es decir, la emisión de un selector 251, que es o bien un bloque de predicción inter emitido por un predictor inter 250 (también conocido como compensador de movimiento) o un bloque de predicción intra emitido por un predictor intra 249). Luego se realiza una
10 transformada directa 242 sobre el bloque residual para emitir un bloque transformado que comprende coeficientes de transformada. Una unidad de cuantificación 243 cuantiza los coeficientes de transformada basándose en un valor de parámetro de cuantificación (QP) (por ejemplo, un valor QP obtenido de un valor de QP de imagen para la imagen en la que el bloque de entrada es parte
15 y un valor de desplazamiento QP específico de bloque para el bloque de entrada), produciendo así coeficientes de transformada cuantificados que luego se codifican en el flujo de bits mediante el codificador 244 (por ejemplo, un codificador aritmético) y el flujo de bits con los coeficientes de transformada codificados se emite desde el codificador 102. A continuación, el codificador 102
20 utiliza los coeficientes de transformada cuantificada para emitir un bloque reconstruido. Esto se hace aplicando primero la cuantificación inversa 245 y la transformada inversa 246 a los coeficientes de transformada para emitir un

bloque residual reconstruido y usando un sumador 247 para añadir el bloque de predicción al bloque residual reconstruido, produciendo así el bloque reconstruido, que se almacena en la memoria intermedia de imagen de reconstrucción (RPB) 266. Se aplica filtrado de bucle mediante un filtro de bucle (LF) etapa 267 y la imagen final decodificada se almacena en una memoria intermedia de imagen decodificada (DPB) 268, donde luego puede ser utilizada por el predictor inter 250 para emitir un bloque de predicción inter para la siguiente imagen a procesar. La etapa LF 267 puede incluir cualquier combinación de las siguientes tres subetapas: i) un filtro de desbloqueo, ii) un filtro de desplazamiento adaptativo de muestra (SAO) y iii) un filtro de bucle adaptativo (ALF).

La FIG. 3 ilustra los componentes funcionales del decodificador 104 de acuerdo con algunas realizaciones. Cabe señalar que el decodificador 104 puede implementarse de forma diferente, por lo que se pueden utilizar implementaciones distintas a este ejemplo específico. El Decodificador 104 incluye un analizador de flujo de bits 301 que recibe el flujo de bits (por ejemplo, recupera el flujo de bits de un archivo o lo recibe a través de la red 110) y analiza el flujo para obtener cada unidad NAL en el flujo de bits. Cuando el analizador de flujo de bits 301 obtiene una unidad NAL que comprende valores de muestra de imagen codificada (por ejemplo, los valores codificados de coeficientes de transformada cuantificada), el analizador de flujo de bits 301 proporciona los valores de muestra codificados a un módulo decodificador 361 (por ejemplo, una

función que realiza decodificación aritmética) que decodifica los valores de muestra para obtener valores de muestra decodificados (por ejemplo, los valores de coeficientes de transformada cuantificada). El decodificador 104 también incluye una etapa de reconstrucción 398 en la que los valores de los coeficientes de transformada cuantificada se someten a un procedimiento de cuantificación inversa 362 y a un procedimiento de transformada inversa 363 para emitir un bloque residual. Este bloque residual se introduce en el sumador 364, que suma el bloque residual y un bloque de predicción que se emite desde el selector 390 para formar un bloque reconstruido. El selector 390 selecciona para generar un bloque de predicción inter o un bloque de predicción intra. El bloque reconstruido se almacena en un RPB 365. El bloque de predicción inter se genera por el módulo de predicción inter 350 y el bloque de predicción intra se genera por el módulo de predicción intra 369. Tras la etapa de reconstrucción 398, una etapa de filtro de bucle 367 aplica filtrado de bucle y la imagen final decodificada puede almacenarse en una memoria intermedia de imagen decodificada (DPB) 368 y enviarse al procesador de imagen 105. Las imágenes se almacenan en el DPB por dos razones principales: 1) esperar la emisión de imagen y 2) usarse como referencia al decodificar imágenes futuras.

Como se ha descrito anteriormente, actualmente existe un desafío porque, actualmente, un decodificador conforme debe decodificar todas las porciones de una imagen (es decir, no puede evitar decodificar una o más porciones de la imagen). En otras palabras, la noción de que diferentes porciones

de la imagen tienen diferentes estados (por ejemplo, distinta importancia, distinta prioridad, etc.) no se reconoce en la tecnología existente. Por lo tanto, la tecnología existente no permite que un decodificador conforme

se abstenga de decodificar una porción de la imagen basándose, por ejemplo, en el estado de la porción y la cantidad de recursos (por ejemplo, memoria o potencia de procesamiento) disponibles para el decodificador, la configuración del dispositivo y/o las preferencias.

En consecuencia, esta divulgación prevé, en un aspecto, indicar un estado para diferentes porciones de una imagen. Se propone indicar el estado de una porción de la imagen en el flujo de bits y, basado en el estado, el decodificador puede elegir decodificar o no la porción (por ejemplo, elegir decodificar aritméticamente o no decodificar aritméticamente los datos codificados aritméticamente correspondientes a la primera porción, como los valores residuales cuantificados).

En consecuencia, como se muestra en la FIG. 15, en una realización el analizador de flujo de bits 301 incluye un selector 1501 que, en una realización, determina, para una o más porciones de imagen en el flujo de bits, un estado de la porción (por ejemplo, determina si la imagen tiene un estado maestro, determina el grupo o grupos al que pertenece la porción, etc.) y luego, basándose en el estado (y otra posible información), determina si seleccionar la porción para decodificar (es decir, determina si proporcionar los valores de muestra codificados de la porción al módulo 361). Es decir, si se selecciona una porción

de imagen para decodificar, entonces los valores de muestras codificadas de esa porción de imagen se proporcionan al módulo de decodificación 361; de lo contrario, los valores de muestras codificadas se descartan (es decir, se eliminan o se envían al “sumidero de bits”).

5 El estado puede indicarse para una porción de la imagen basada en la prioridad, importancia, análisis del contenido u otras preferencias relacionadas con esa porción de la imagen. Por lo tanto, un primer decodificador puede decodificar solo aquellas porciones de imagen indicadas con estado A, un segundo decodificador puede decodificar solo aquellas porciones de imagen
10 indicadas con estado B, y un tercer decodificador puede decodificar solo aquellas porciones de imagen indicadas con estado A o B.

 En una realización, el flujo de bits codificado incluye varias subimágenes por imagen (cada subimagen comprende una o más porciones de la imagen) y al menos una de estas subimágenes se indica que tiene un estado “maestro” (es
15 decir, la subimagen es una subimagen maestra). En una realización, una subimagen que tiene estado maestro significa que la subimagen incluye contenido con mayor importancia para el usuario en comparación con otras subimágenes en la imagen. Un decodificador con recursos limitados puede entonces decodificar solo una o más porciones de la imagen correspondiente a
20 la subimagen maestra y no decodifican otras porciones que no forman parte de una subimagen maestra.

Realización 1: Un procedimiento para decodificar una imagen

En una realización se proporciona un procedimiento para decodificar una imagen de un flujo de bits, en el que la imagen comprende una primera y una segunda porción. En una realización, la primera porción es “independiente” al menos de la segunda porción de la imagen. Como se usa aquí, una porción de una imagen es “independiente” de otra porción de la imagen cuando la porción puede reconstruirse correctamente sin ninguna información de la otra porción (es decir, la porción independiente no predice de la otra porción). El procedimiento comprende obtener, del flujo de bits, la primera información de estado (por ejemplo, un primer valor indicador de estado que indica un estado, como, por ejemplo, un nivel de prioridad) correspondiente a (es decir, relacionado con) la primera porción.

El procedimiento incluye además, determinar basándose en la primera información de estado correspondiente a la primera porción, si decodificar o no la primera porción. Por ejemplo, en una realización, determinar si decodificar o no la primera porción basándose en la primera información de estado comprende determinar si la primera información de estado es igual a un primer valor y, si la primera información de estado es igual al primer valor, entonces se determina decodificar la primera porción de la imagen. En una realización, si la primera información de estado no es igual al primer valor, se determina no decodificar la primera porción (es decir, abstenerse de decodificar la primera porción). En otra realización, si la primera información de estado no es igual al primer valor,

entonces la determinación de si decodificar o no la primera porción puede depender de que se cumplan otros criterios (por ejemplo, que el decodificador tenga recursos suficientes).

El primer valor puede ser un valor que está codificado de forma fija en el decodificador o que el decodificador obtiene desde un archivo de configuración, del flujo de bits o de la entrada del usuario.

En una realización, la primera información de estado corresponde a la primera porción y no a la segunda porción. En una realización, una porción de la imagen es una subimagen, un segmento, una tesela, una CTU, un conjunto de CTU, un conjunto de teselas o un grupo de teselas.

En una realización, la información de estado es el valor de un indicador y se decodifica de un elemento sintáctico en el flujo de bits. En una realización, el indicador puede por ejemplo ser un indicador binario donde la información de estado es uno de dos valores posibles. En otra realización, el indicador puede codificar uno de tres o más valores.

En esta divulgación, un estado para una porción de una imagen puede definirse como un valor específico del indicador asociado a esa porción señalizada en el flujo de bits. Indicar un estado para una porción de una imagen puede especificar que se invocará un procedimiento en la decodificación de esa porción o de toda la imagen. El decodificador puede elegir etapas alternativas para manejar porciones independientes de una imagen basada en el estado de cada porción. Un ejemplo es un valor indicador en el flujo de bits que indica que

una primera porción de una imagen tiene el estado “maestro” y el procedimiento de decodificación para decodificar esta imagen comprende decodificar la primera porción en respuesta a que esta porción tiene el estado “maestro”. Una porción de una imagen que tiene estado “maestro” puede interpretarse como que esa porción es preferida (por ejemplo, priorizada) frente a otras porciones de la imagen que no tienen estado “maestro”, por ejemplo, en caso de recursos limitados para decodificar. Otra interpretación puede ser que la porción de la imagen que tenga el estado “maestro” debe ser decodificada, mientras que otras porciones de la imagen que no tienen estado “maestro” no deben ser o pueden no ser decodificadas, y para las cuales el flujo de bits puede ni siquiera comprender ningún dato de la capa de codificación de vídeo.

Una porción de la imagen con estado “maestro” puede definirse como una porción de la imagen que se decodifica en todas las versiones de los decodificadores. Esto significa que la decodificación de la porción no es opcional, y que está garantizado que cualquier decodificador compatible que funcione correctamente debe decodificar (o al menos intentar decodificar) la porción con estado “maestro”.

El estado también puede verse como un agrupamiento de la una o más porciones de la imagen. En un ejemplo, hay al menos dos porciones en la imagen y se ha asignado un estado A o un estado B a cada porción de la imagen donde al menos una porción tiene estado A y al menos una porción tiene estado B. En este ejemplo se puede decir que la imagen está dividida en dos grupos de

porciones de imagen: un grupo son porciones con estado A y otro grupo son porciones con estado B. En un ejemplo, el decodificador decodifica porciones con estado A y no decodifica, o puede que no decodifique, porciones con estado B en la imagen.

5 Los estados pueden aportar ventaja al arte actual agrupando porciones de acuerdo con sus propiedades, de modo que algunas porciones de la imagen que comparten una propiedad particular (por ejemplo, porciones que pertenecen a la misma vista particular) se decodifican mientras que otras porciones de la imagen que no tienen esa propiedad particular (por ejemplo, no pertenecen a la vista particular) pueden no ser decodificadas dependiendo de otros factores (por
10 ejemplo, preferencias del usuario, recursos disponibles, etc.). Las propiedades pueden incluir las siguientes:

(a) ID de vista de porción, por ejemplo, un estado indicaría un ID de vista de porción particular, donde una o más porciones de la imagen que tiene un
15 estado particular comprenderían datos para una vista particular en una imagen, donde una vista puede ser una vista en una configuración multivista, una vista capturada en un ángulo específico de un contenido determinado, como una región de interés (ROI), o una vista de un contenido específico entre múltiples contenidos, y/o

20 (b) Tipo de datos, por ejemplo, textura, alfa, profundidad, atributo o metadatos de nube de puntos, etc. Por ejemplo, un valor de estado podría

corresponder a la(s) porción(es) de imagen asociada(s) que comprenda una de datos de textura, de alfa o de profundidad, y/o

(c) Prioridad de los datos de vídeo a usar para análisis de vídeo. Por ejemplo, algunos objetos en un vídeo pueden ser más importantes de rastrear
5 que otros.

En un ejemplo, la primera porción es independiente de otras porciones y es una primera subimagen. En este ejemplo, se decodifica un valor indicador igual a un primer valor “estado de subimagen maestra” y, en respuesta a que el valor indicador es igual al estado de subimagen maestra, se decodifica la primera
10 subimagen.

Como se usa aquí, la frase “una porción de la imagen que tiene un primer estado” se usa indistintamente con la frase “una porción de primer estado”. Como un ejemplo, “una subimagen que tiene estado de subimagen maestra” es lo mismo que “una subimagen maestra” y “una subimagen con estado maestro”.

15 En otro ejemplo, la imagen se divide al menos en una primera y una segunda porción. En este ejemplo, un primer estado para la primera porción se indica con un primer valor indicador en el flujo de bits para la primera porción y un segundo estado para la segunda porción se indica con un segundo valor indicador en el flujo de bits para la segunda porción. Los valores indicadores para
20 el primer estado y el segundo estado pueden señalizarse en el mismo elemento sintáctico o en diferentes elementos sintácticos. El procedimiento comprende decodificar la primera porción de la imagen en respuesta a que el primer valor

indicador sea igual a un primer valor o que el primer estado sea igual a un estado A, y no a decodificar la primera porción de la imagen en respuesta a que el primer valor indicador sea igual a un segundo valor o que el primer estado sea igual a un estado B, diferente del estado A. De igual modo, la segunda porción de la
5 imagen se decodifica en respuesta a que el segundo valor indicador es igual al primer valor o el segundo estado es igual a un estado A, y la segunda porción de la imagen no se decodifica en respuesta a que el segundo valor indicador sea igual a un segundo valor o que el segundo estado sea igual a estado B.

En una realización, la primera porción de la imagen es una primera
10 subimagen en la imagen, la segunda porción de la imagen es una segunda subimagen en la imagen, el estado A significa el estado de la subimagen maestra, y el estado B significa el estado de la subimagen no maestra. Por ejemplo, cuando el primer estado correspondiente a la primera porción es igual al estado A y el segundo estado correspondiente a la segunda subimagen es
15 igual al estado B, el estado de la primera subimagen es un estado de subimagen maestra, y el estado de la segunda subimagen es un estado de subimagen no maestra.

En una realización, la primera subimagen de la imagen se decodifica en respuesta a determinar que la primera subimagen de la imagen tiene un estado
20 A donde el estado A es un estado de subimagen maestra. En una realización, la segunda subimagen de la imagen no se decodifica como resultado de determinar que la segunda subimagen tiene un estado B, donde el estado B es el estado de

la subimagen no maestra. En otra realización, la segunda subimagen no se decodifica como resultado de determinar que la segunda subimagen tiene un estado de “B” y que satisfaga la determinación de una condición (por ejemplo, determinar que la cantidad de memoria disponible y/o de recursos de procesamiento disponibles están por debajo de un umbral y/o determinar que no se cumplirá un requisito de latencia si se intenta decodificar la segunda subimagen).

El dispositivo 103 puede realizar todos o un subconjunto de las siguientes etapas para decodificar una imagen de un flujo de bits:

10 (1) Obtener el flujo de bits (por ejemplo, recibir el flujo de bits a través de la red 110 o recuperar el flujo de bits de una unidad de almacenamiento), donde el flujo de bits contiene información para decodificar una imagen actual del flujo de bits;

15 (2) Decodificación de información del flujo de bits que indica que la imagen actual comprende una primera porción (la información también puede indicar que la primera porción es independiente de otras porciones de la imagen);

 (3) Obtener un primer valor indicador;

20 (4) Determinar si decodificar o no la primera porción basándose en el primer valor indicador (por ejemplo, decodificar la primera porción en respuesta a que el primer valor indicador sea igual a un primer valor, y no decodificar la primera porción en respuesta a que el primer valor indicador no sea igual al primer valor).

En otra realización, el dispositivo 103 puede realizar todos o un subconjunto de las siguientes etapas para decodificar una imagen de un flujo de bits:

- 5 (1) Obtener un flujo de bits que contenga información para decodificar una imagen actual del flujo de bits;
- (2) Decodificar información del flujo de bits de que la imagen actual está particionada en, al menos, una primera porción y una segunda porción;
- (3) Obtener un primer valor indicador para la primera porción (el primer valor indicador puede decodificarse del flujo de bits);
- 10 (4) Determinar un primer estado del primer valor indicador;
- (5) Decodificar la primera porción en respuesta a que el primer valor indicador sea igual a un primer valor o que el primer estado sea igual al estado A;
- (6) No decodificar la primera porción en respuesta a que el primer valor
15 indicador sea igual a un segundo valor o que el primer estado sea igual al estado B;
- (7) Obtener un segundo valor indicador para la segunda porción (el segundo valor indicador puede ser decodificado del flujo de bits);
- (8) Decodificar la segunda porción en respuesta a que el segundo valor
20 indicador sea igual a un primer valor o que el segundo estado sea igual al estado A;

(9) No decodificar la segunda porción en respuesta a que el segundo valor indicador sea igual a un segundo valor o el segundo estado sea igual al estado B.

En un ejemplo, una porción de una imagen debe tener un primer o
5 segundo estado y la primera porción de la imagen tiene un primer estado, y la segunda porción de la imagen no tiene el primer estado. Esto puede interpretarse como que la segunda porción de la imagen tiene un segundo estado que no es igual al primero.

En otro ejemplo, una imagen tiene más de una porción, cada porción de
10 la imagen tiene un primer estado o algún estado diferente al primer estado (es decir, algunas porciones pueden tener el primer estado mientras otras porciones no tienen el primer estado), solo las porciones con el primer estado están expresamente indicadas de modo que cualquier porción que no se indique expresamente como teniendo el primer estado necesariamente tiene otro
15 estados distinto al primero estado (esto es, se entiende implícitamente que las demás porciones de la imagen que no han sido indicadas con el primer estado no tienen dicho primer estado). Por ejemplo, en una realización hay al menos una porción en la imagen con un estado A y al menos una porción en la imagen con un estado B, y el estado A y el estado B no son iguales.

20 En una realización, el primer valor indicador se decodifica de un conjunto de parámetros, un encabezado o un mensaje SEI. Un encabezado puede, por

ejemplo, ser un encabezado de imagen, un encabezado de segmento o un encabezado de unidad NAL.

En una realización, la primera porción comprende dos o más CTU.

El codificador 102 puede realizar todos o un subconjunto de las siguientes
5 etapas para codificar una imagen en un flujo de bits:

(1) Determinar que la imagen comprende una o más porciones;

(2) Determinar que una primera porción de la imagen está indicada por un
valor indicador, en el que el valor indicador igual a un primer valor indica que la
primera porción de la imagen debe ser decodificada por el decodificador y: (2a)
10 el valor indicador no igual al primer valor indica que la primera porción de la
imagen puede o no ser decodificada por el decodificador; o (2b) el valor indicador
igual a un segundo valor indica que la primera porción de la imagen puede o no
ser decodificada por el decodificador;

(3) Codificar el valor indicador de la primera porción de la imagen en el
15 flujo de bits;

(4) Si el valor indicador es igual al primer valor, codificar la primera porción
de la imagen en el flujo de bits;

(5) Si el valor indicador no es igual al primer valor o es igual al segundo
valor, codificar la primera porción de la imagen en el flujo de bits o abstenerse
20 de codificar la primera porción de la imagen.

En otra realización, el codificador 102 puede realizar todos o un subconjunto de las siguientes etapas para codificar una imagen en un flujo de bits:

(1) Determinar que la imagen comprende al menos una primera porción y
5 una segunda porción;

(2) Codificar un primer valor indicador para la primera porción de la imagen en el flujo de bits, donde el valor del indicador igual al primer valor indica que la primera porción de la imagen debe ser decodificada por el decodificador, y que el primer valor indicador no sea igual al primer valor indica que la primera porción
10 de la imagen puede o no ser decodificada por el decodificador;

(3) Si el primer valor indicador es igual al primer valor, codificar la primera porción de la imagen en el flujo de bits;

(4) Si el primer valor indicador no es igual al primer valor, codificar la primera porción de la imagen en el flujo de bits o abstenerse de codificar la
15 primera porción de la imagen en el flujo de bits;

(5) Codificar un segundo valor indicador para la segunda porción de la imagen en el flujo de bits, donde el segundo valor indicador igual al primer valor indica que la segunda porción de la imagen debe ser decodificada por el decodificador, y que el segundo valor indicador no sea igual al primer valor indica
20 que la segunda porción de la imagen puede o no ser decodificada por el decodificador;

(6) Si el segundo valor indicador es igual al primero, codificar la segunda porción de la imagen en el flujo de bits;

(7) Si el segundo valor indicador no es igual al primer valor, codificar la segunda porción de la imagen en el flujo de bits o abstenerse de codificar la
5 segunda porción de la imagen en el flujo de bits.

El dispositivo 103 puede, en este ejemplo, realizar todos o un subconjunto de las siguientes etapas para decodificar una imagen que comprende una primera porción y una segunda porción de un flujo de bits, en los que las porciones comprendidas en la imagen son independientes:

10 (1) Decodificar un primer valor del flujo de bits, el primer valor que especifica que la primera porción tiene un estado A;

(2) Decodificar un segundo valor del flujo de bits, siendo el segundo valor diferente del primer valor y el segundo valor especificando que la segunda porción tiene un estado B distinto al estado A;

15 (3) Decodificar la primera porción en respuesta a que la primera porción tiene estado A; y

(4) No decodificar la segunda porción en respuesta a que la segunda porción tiene estado B.

20 Realización 2: Acciones extra para rellenar valores de muestra sin decodificación

Esta realización describe acciones adicionales que el decodificador puede realizar para decodificar una imagen de un flujo de bits. En esta realización, el

decodificador no decodifica la primera porción en respuesta a que el primer valor indicador no sea igual al primer valor y, además, el decodificador puede:

(1) Generar valores de muestra para la primera porción de la imagen usando, por ejemplo: i) un valor de muestra constante, ii) valores de muestra por defecto y/o di) valores de muestra previamente decodificados; y/o

(2) Copiar, interpolar o extrapolar valores de muestra para la primera porción de la imagen desde: i) valores de muestra colocados en una o más imágenes previamente decodificadas, ni) otros valores de muestra en una o más imágenes previamente decodificadas, ni) otros valores de muestra en la imagen actual, una imagen predeterminada y/o parte de una imagen, y el decodificador rellena los valores de muestra en la primera porción con los valores de muestra generados, copiados, interpolados y/o extrapolados.

En un ejemplo, dos imágenes P1 y P2 con los mismos valores de muestra excepto en una primera porción se codifican en flujo de bits B1 y flujo de bits B2, respectivamente, y la primera porción en P1 y la primera porción en P2 están indicadas para no ser decodificadas por el decodificador. Usando esta realización, el decodificador ignora las porciones del flujo de bits B1 y B2 que representan la primera porción; sin embargo, las imágenes decodificadas P'1 y P'2 tendrán los mismos valores de muestra en todas partes, ya que los valores codificados correspondientes a la primera porción se ignoran, pero el procedimiento de relleno es el mismo y por lo tanto la primera porción de P'1 y P'2 se rellena con los mismos valores.

En un ejemplo, el decodificador ha determinado que la primera subimagen no es una subimagen maestra y, en respuesta a ello, el decodificador no decodifica la primera subimagen y, además, copia los valores de muestra de un área de colocación de una imagen previamente decodificada en los valores de muestra de la primera subimagen de la imagen actual.

En una variante de esta realización, se señala en el flujo de bits cómo generar, copiar, interpolar o extrapolar los valores de muestra para las porciones de la imagen que no están decodificadas. De acuerdo con esta realización, el dispositivo 103 puede realizar todos o un subconjunto de las siguientes etapas mostradas en la Tabla 3 para decodificar una imagen de un flujo de bits:

TABLA 3

1.	Recibir un flujo de bits que contiene información para decodificar una imagen actual del flujo de bits
2.	Decodificar del flujo de bits que la imagen actual está particionada en al menos una primera y una segunda porción
3.	Decodificar un valor indicador para la primera porción
4.	Determinar un estado del valor indicador
5.	Decodificar la primera porción en respuesta a que el valor indicador es igual a un primer valor o el estado es un primer estado
6.	Siguiendo las siguientes etapas en respuesta a que el valor indicador sea igual a un segundo valor o que el estado sea un segundo estado:
a.	No decodificar la primera porción

- b. Decodificar un segundo indicador del flujo de bits
- c. Generar o copiar valores de muestra para la primera porción de la imagen usando el segundo valor indicador
- d. Rellenar los valores de muestra de la primera porción de la imagen con los valores de muestra generados o copiados

En un ejemplo, un valor indicador en el flujo de bits igual a un primer valor indica que el decodificador no decodifica porciones de la imagen que no tienen estado maestro y el decodificador rellena los valores de muestra en todas las
5 porciones de la imagen que no son porciones maestras con gris. El valor indicador igual a un segundo valor en este ejemplo indica que el decodificador rellena los valores de muestra en todas las porciones de la imagen que no son porciones maestras con negro.

En un ejemplo, la información para decodificar valores de muestra para la
10 parte que no es parte maestra ni siquiera existe en el flujo de bits.

Realización 3: Cambio de disposición de subimagen en la imagen decodificada o emitida.

En esta realización, el decodificador reorganiza la posición y/o reescala una porción de la imagen. En esta realización, una primera porción de una
15 imagen se decodifica en respuesta a determinar que la primera porción de la imagen tiene un primer estado, y una segunda porción de la imagen no se decodifica en respuesta a determinar que la segunda porción de la imagen tiene

un segundo estado, y además el decodificador puede: (1) reubicar al menos una de las porciones de la imagen; (2) escalar o reescalar al menos una de las porciones de la imagen; y/o (3) cambiar la resolución de emisión de la imagen decodificada o de una porción decodificada de la imagen.

5 En un ejemplo, el decodificador decodifica una primera subimagen en la imagen en respuesta a que la primera subimagen tiene un estado de subimagen maestra y el decodificador no decodifica una segunda subimagen en la imagen en respuesta a que la segunda subimagen no tenga estado de subimagen maestra, y, además, el decodificador escala la primera subimagen para cubrir
10 toda el área de la imagen antes de emitir la imagen.

 En una versión de esta realización, la selección de subimágenes que se decodifican y se emiten se basa en la orientación del dispositivo de visualización, donde el dispositivo de visualización puede ser, por ejemplo, un teléfono o un televisor. En un caso de uso de ejemplo, una imagen 400 (véase la FIG. 4A)
15 consiste de 9 subimágenes (numeradas del 0 al 8 en la FIG. 4A) y la imagen es más grande que la que se muestra en cualquier pantalla. Si se muestran verticalmente, solo se muestran las subimágenes 1, 4 y 7, y si se muestran horizontalmente solo las subimágenes 3, 4 y 5. En este ejemplo, las subimágenes 1, 4 y 7 podían indicarse como subimágenes maestras para una
20 pantalla vertical y las subimágenes 3, 4 y 5 podían indicarse como subimágenes maestras para una pantalla horizontal. Las subimágenes 0, 2, 6 y 8 nunca se muestran y en una versión no hay datos VCL en el flujo de bits para las

subimágenes 0, 2, 6 y 8. En otra versión, solo la subimagen 4 tiene un estado de subimagen maestra y las subimágenes 1, 7 tienen un estado X y las subimágenes 3 y 5 tienen un estado Y. Un decodificador puede entonces seleccionar o recibir instrucciones para decodificar y emitir la subimagen maestra y las subimágenes con estado X o las subimágenes con estado Y, dependiendo, por ejemplo, de la orientación de la pantalla.

Esto puede hacerse sin subimágenes, pero la ventaja podría ser que las subimágenes que no se mostrarán no necesitan ser decodificadas. A menudo está bien esperar un segundo si la orientación de la pantalla está invertida.

De acuerdo con esta realización, el dispositivo 103 puede realizar todos o un subconjunto de las etapas mostradas en la Tabla 4 para decodificar una imagen de un flujo de bits:

TABLA 4

1.	Obtener un flujo de bits que contenga información para decodificar una imagen actual del flujo de bits
2.	Decodificar del flujo de bits que la imagen actual está particionada en al menos una primera y una segunda porción
3.	Decodificar del flujo de bits una primera información sobre la ubicación y/o posición de la primera y/o segunda porción.
4.	Decodificar un primer valor indicador del flujo de bits para la primera porción y un segundo valor indicador para la segunda porción

5. Determinar un primer estado para la primera porción basándose en el primer valor indicador y un segundo estado para la segunda porción basado en el segundo valor indicador
6. Decodificar la primera porción en respuesta al primer estado que es indicado para la primera porción
7. No decodificar la segunda porción en respuesta al segundo estado que es indicado para la segunda porción.
8. Reescalar, reorganizar, redimensionar y/o cambiar la relación de aspecto de la primera porción para que la ubicación, posición, tamaño y/o relación de aspecto de la primera porción en la imagen de emisión sea diferente de la ubicación, posición, tamaño y/o relación de aspecto de la primera porción especificada por la primera información en el flujo de bits. La nueva ubicación, posición, tamaño y/o relación de aspecto de la primera porción puede especificarse en el flujo de bits o puede ser independiente del flujo de bits y determinarse en una etapa puramente de postprocesamiento.
9. Emitir la imagen decodificada

Realización 4: Control de nivel de secuencia al permitir subimágenes maestras

- En esta realización, se indica en el flujo de bits (por ejemplo, mediante un elemento de sintaxis de control) si se permite decodificar o no una porción de una imagen basándose en un valor indicador señalizado en el flujo de bits o no.
- 5

En una variante, un valor de un elemento sintáctico de control decodificado significa que todas las porciones deben ser decodificadas independientemente de sus valores de estado decodificados. Otro valor de un elemento sintáctico de control decodificado significa que las porciones que tienen un cierto estado no pueden o no deben ser decodificadas. En otra variante, un valor de un elemento de sintaxis de control decodificado significa que no hay elementos de sintaxis de valor de estado en el flujo de bits (es decir, la presencia de elementos de sintaxis de valor de estado en el flujo de bits está condicionada por el elemento de sintaxis de control).

En una variante de esta realización, se indica en el flujo de bits si se permite indicar un estado para una porción de una imagen en el flujo de bits o no. En una variante, un valor de un elemento de sintaxis de control decodificado significa que un determinado valor de estado no debe ser señalizado en ningún elemento de sintaxis de valor de estado en el flujo de bits. Se pueden permitir valores de estado distintos a cierto valor. Otro valor de un elemento de sintaxis de control decodificado significa que ese determinado valor de estado puede ser señalizado por elementos de sintaxis de valor de estado en el flujo de bits.

El elemento de sintaxis de control decodificado puede ser un elemento sintáctico a nivel de secuencia, y puede estar ubicado en un conjunto de parámetros como un conjunto de parámetros de vídeo o conjunto de parámetros de secuencia.

En una realización, un tercer indicador se decodifica del flujo de bits y el valor del tercer indicador, siendo igual a un tercer valor, especifica que el primer indicador para al menos una primera porción de la imagen puede estar presente en el flujo de bits y el valor del tercer indicador es igual a un cuarto valor, 5 especifica que el primer indicador para al menos una primera porción de la imagen no está presente en el flujo de bits.

En una realización, una indicación a nivel de secuencia especifica si se permite el uso de subimágenes maestras.

En otro ejemplo, la indicación es un indicador en el SPS, donde el 10 indicador igual a 1 especifica que se pueden indicar subimágenes maestras en el flujo de bits y el indicador igual a 0 especifica que las subimágenes maestras no deben indicarse en el flujo de bits.

Realización 5: Información de disposición

En esta realización, el flujo de bits contiene i) información de primera 15 disposición (o una “primera disposición” para abreviar) donde la primera disposición especifica un primer conjunto de porciones de imagen e ii) información de segunda disposición (o una “segunda disposición” para abreviar), en el que la segunda disposición especifica al menos un segundo conjunto de porciones de imagen donde al menos un estado está indicado para al menos una 20 de las porciones del segundo conjunto.

En un ejemplo, la primera disposición señalizada en el flujo de bits especifica, para cada porción de la imagen, las coordenadas x-y de una esquina

de la porción y las dimensiones de la porción (por ejemplo, ancho y alto). Las coordenadas x-y pueden estar en unidades de tesela o en unidades de bloque.

En un ejemplo, la segunda disposición especifica cómo se agrupan las porciones de imagen del primer conjunto de acuerdo con su estado, en el que
5 las porciones de cada grupo tienen el mismo estado. En una versión preferida de esta realización, la primera y la segunda disposición difieren.

La FIG. 4B ilustra un ejemplo de un primer conjunto de porciones (es decir, las porciones 0, 1, 2 y 3) que se especifican con la primera información de disposición y un segundo conjunto de porciones (es decir, las porciones 0 y 1)
10 que se especifican con la segunda información de disposición. En una realización, cada porción que es miembro del segundo conjunto de porciones se considera que tiene un cierto estado (por ejemplo, estados maestro) simplemente por el hecho de que la porción es miembro del segundo conjunto. Pero en otra realización, la segunda disposición no solo especifica expresamente
15 qué porciones pertenecen al segundo conjunto de porciones, sino que también especifica expresamente el estados asignado al segundo conjunto de porciones (por ejemplo, estado maestro), y, por lo tanto, implícitamente asignado a cada porción que es miembro del segundo conjunto de porciones.

Es decir, por ejemplo, en una realización específica en la que las
20 porciones son subimágenes, la primera disposición especifica la partición de subimágenes y la segunda disposición especifica un grupo de subimágenes con el estado maestro. Esto se ilustra en la FIG. 4B, donde la primera disposición

especifica la partición de la imagen en subimágenes 0 a 3 y la segunda disposición especifica las subimágenes 0 y 1 como el grupo de subimágenes que tienen estado maestro.

En una realización, la primera disposición y la segunda disposición se decodifican del flujo de bits. En una realización, decodificar la primera disposición comprende decodificar una palabra código que especifica el número total de porciones de imagen y luego decodificar, para cada porción, un conjunto de información que especifica la porción (por ejemplo, coordenadas x-y, dimensiones e ID). La Tabla 5 muestra una sintaxis ejemplar para la primera disposición.

TABLA 5

Sintaxis ejemplar para identificar porciones de imagen

parameter_set_rbsp() {	Descriptor
...	
num_portions_minus1	u(8)
for(i = 0; i <= num_portions_minus1; i++) {	
portion_ctu_top_left_x[i]	u(v)
portion_ctu_top_left_y[i]	u(v)
portion_width_minus1[i]	u(v)
portion_height_minus1[i]	u(v)
portion_ID[i]	u(v)

...	
}	
...	

En una realización, decodificar la segunda disposición comprende decodificar una palabra código que especifica el tamaño del segundo conjunto de porciones (es decir, el número total de porciones incluidas en el segundo conjunto), y luego decodificar palabras código número S, donde cada una de las palabras código S identifica una porción que es miembro del segundo conjunto de porciones. La tabla 6A muestra una sintaxis ejemplar para la segunda disposición:

TABLA 6A

10 Sintaxis ejemplar para especificar las porciones que pertenecen al segundo conjunto

parameter_set_rbsp() {	Descriptor
...	
S_minus1	u(8)
for(i = 0; i <= S_minus1; i++) {	
second_set [i]	u(v)
}	
...	

Es decir, el arreglo “second_set” tendrá elementos S. En una realización, cada elemento del segundo arreglo second_set (que también puede llamarse arreglo “master_set”) incluirá un ID de porción. De este modo se pueden identificar las porciones que pertenecen al segundo conjunto.

5 La tabla 6B muestra otra sintaxis ejemplar para la segunda disposición:

TABLA 6B

Sintaxis ejemplar para especificar las porciones que pertenecen al segundo conjunto

parameter_set_rbsp() {	Descriptor
...	
second_set	u(v)
...	

10 En la realización mostrada en la Tabla 6B, la variable second_set es una máscara de bits (es decir, una cadena de bits) que identifica las porciones que son miembros del segundo conjunto. Por ejemplo, si el número total de porciones incluidas en el primer conjunto es N, la máscara de bits tendrá una longitud de N donde cada bit de la máscara de bits corresponde a una de las N porciones, y,

15 si el bit de una porción está fijado en 1, entonces la porción es miembro del second_set, de lo contrario, no es miembro del segundo conjunto. Como ejemplo específico, si el second_set = 0101 entonces las porciones 0 y 2 son miembros

del segundo conjunto, y si el `second_set = 1000`, entonces solo la porción 3 es miembro del segundo conjunto.

Alternativamente, se indica para cada porción incluida en el primer conjunto de porciones si la porción pertenece o no al segundo conjunto de porciones. En esta alternativa, el número de porciones incluidas en el segundo conjunto no necesita indicarse con un elemento sintáctico dedicado, sino derivarse de las indicaciones para cada porción. La Tabla 7 muestra una sintaxis ejemplar para esta realización.

TABLA 7

<code>parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>num_portions_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_portions_minus1; i++) {</code>	
<code>portion_ctu_top_left_x[i]</code>	<code>u(v)</code>
<code>portion_ctu_top_left_y[i]</code>	<code>u(v)</code>
<code>portion_width_minus1[i]</code>	<code>u(v)</code>
<code>portion_height_minus1[i]</code>	<code>u(v)</code>
<code>portion_ID[i]</code>	<code>u(v)</code>
<code>portion_member_of_second_set_flag[i]</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	

...	
-----	--

En otra realización, una primera porción incluida en el primer conjunto de porciones se incluye en un segundo conjunto de porciones (también conocido como segundo grupo o estado A) y una segunda porción que se incluye en el primer conjunto se incluye en un tercer conjunto de porciones (también conocido como tercer grupo o estado B). En tal realización, decodificar la segunda disposición comprende decodificar una primera palabra código que especifica un número total de grupos (NG) (es decir, el número total de estados distintos) y, para cada grupo G_i , $0 \leq i \leq NG-1$ en la segunda disposición, se identifican las porciones de la primera disposición que se incluyen en G_i . La Tabla 8 muestra una sintaxis ejemplar para esta realización.

TABLA 8

parameter_set_rbsp() {	Descriptor
...	
NG_minus1	u(8)
para (i = 0; i <= NG_minus1; i++) {	
group_members[i]	u(v)
group_status[i]	u(v)
}	
...	

Como se muestra la Tabla 8, el arreglo "group_members" tendrá elementos NG. En una realización, cada elemento del arreglo group_members incluirá una máscara de bits (es decir, una cadena de bits) que identifica las porciones que son un miembro del grupo. Por ejemplo, si el número total de porciones incluidas en el primer conjunto es N, entonces cada elemento en el

5 porciones incluidas en el primer conjunto es N, entonces cada elemento en el arreglo group_members será una máscara de bits de N bits donde cada bit corresponde a una de las N porciones y, si el bit de una porción está fijado en 1, entonces la porción es miembro del grupo, de lo contrario, no es miembro del grupo. Como ejemplo concreto, si group_members[0] = 0101 entonces las

10 porciones 0 y 2 son un miembro del grupo 0, y si group_members[l] = 1000, entonces solo la porción 3 es un miembro del grupo 1.

En otra realización, cada elemento del arreglo group_members incluirá una lista de los ID de porción (como se muestra en la Tabla 7, a cada porción se le da un único ID de porción). Es decir, la segunda disposición utiliza los ID de

15 porción que son diferentes del índice de porción pero únicos para cada porción para indicar los agrupamientos. En un ejemplo, el índice de porción se especifica con el índice i en la Tabla 5 anterior, como el valor del contador de bucle para cada porción en ese orden particular, así que la primera porción obtiene el índice 0 y la siguiente porción obtiene el índice 1, etc. El ID de la porción se señala

20 entonces en el flujo de bits (por ejemplo, como ID de la porción) para crear un mapeo entre cada índice de porción en un valor de ID único A. El beneficio de este procedimiento podría ser que una actualización de estado es posible

actualizando solo el mapeo entre el índice y los ID, sin necesidad de reenviar los valores de partición de porciones o valores de estado. En un ejemplo de esta variante, hay una correspondencia uno a uno de un índice de porción en un ID de porción y la indicación se señala en un bucle sobre los ID de porción.

- 5 Además, como se muestra en la Tabla 8, el flujo de bits puede incluir, para cada uno de los grupos NG, una palabra código que contiene información que indica el estado asignado al grupo (aunque esto no es necesario en la realización donde cada grupo tiene un estado implícito). En consecuencia, un decodificador puede obtener del flujo de bits: información de la porción de imagen que
- 10 especifica un conjunto de porciones de imagen, información de pertenencia a grupos que indica el número de grupos de porciones de imagen y, para cada grupo de porciones de imagen, los miembros del grupo y el estado asignado al grupo, si es necesario.

- Alternativamente, se indica para cada porción P1 de la primera disposición
- 15 a qué grupo pertenece P1 en la segunda porción. En esta alternativa, el número de grupos que hay en la segunda disposición puede no estar indicado por un elemento sintáctico dedicado, sino derivarse de las indicaciones para cada porción P1. La Tabla 9 muestra una sintaxis ejemplar para esta realización.

TABLA 9

parameter_set_rbsp() {	Descriptor
...	
num_portions_minus1	u(8)

for(i = 0; i <= num_portions_minus1; i++) {	
portion_ctu_top_left_x[i]	u(v)
portion_ctu_top_left_y[i]	u(v)
portion_width_minus1[i]	u(v)
portion_height_minus1[i]	u(v)
portion_groupID[i]	u(v)
...	
}	
...	

En el ejemplo anterior, la segunda disposición se señala en el mismo bucle que la primera disposición.

La FIG. 5 ilustra un ejemplo donde dos disposiciones diferentes de subimagen usando la misma disposición de subimagen maestra indican dos porciones distintas de la imagen con estado maestro (las porciones con estados maestro son las porciones dentro del rectángulo punteado). Una ventaja de esta realización es que para actualizar el área de la imagen que tiene el estado A basta con actualizar las porciones de la imagen que están en el grupo especificado con el estado A, y no es necesario modificar la partición de la imagen en porciones.

La FIG. 6 ilustra un ejemplo de cómo una actualización en la disposición de la subimagen maestra afecta a la porción de la imagen con estado maestro.

En un ejemplo, una disposición de subimagen se señala en el flujo de bits y un indicador o un elemento sintáctico índice en un bucle de un encabezado o conjunto de parámetros (por ejemplo, el PH o el PPS) especifica cuáles de las subimágenes son subimágenes maestras. En una variante, un elemento de sintaxis índice en el SPS, PPS o PH especifica cuáles de las subimágenes son subimágenes maestras. Un ejemplo similar utiliza segmentos como porciones de la imagen.

En un ejemplo, en el SPS se señalizan una disposición de subimagen y una disposición de subimagen maestra.

En otro ejemplo, una disposición de subimagen se señala en el SPS y una disposición de subimagen maestra (también conocida como información de identificación de subimagen maestra) se señala en el PPS. En este ejemplo, la disposición de la subimagen maestra puede actualizarse en el flujo de bits señalizando un nuevo PPS con una nueva disposición de subimagen maestra (es decir, nueva información que identifica qué porciones son porciones maestras) sin necesidad de actualizar la disposición de la subimagen en el SPS. En otro ejemplo, la disposición de la subimagen se señala en el SPS o PPS y la subimagen maestra se señala en un encabezado de imagen o encabezado de segmento. En este ejemplo, la disposición de la subimagen maestra puede actualizarse para cada imagen o segmento.

En otro ejemplo, la segunda disposición está señalizada en el mismo bucle que la primera disposición. Por ejemplo, una indicación de que la i -ésima

subimagen es la subimagen maestra se señala en el mismo bucle donde se señala la disposición de la subimagen. En un ejemplo, hay un indicador para cada subimagen en la disposición de la subimagen SPS que especifica si esta subimagen es una subimagen maestra o no. La Tabla 10 muestra una sintaxis

5 ejemplar para indicar si la i-ésima subimagen es una subimagen maestra usando el indicador `sps_master_subpic_flag[ i ]` en el mismo bucle donde se señala la disposición de las subimágenes.

TABLA 10

Sintaxis ejemplar para indicar si la i-ésima subimagen es una subimagen

10 maestra en el mismo bucle donde se señala la disposición de las subimágenes.

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>sps_num_subpics_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= sps_num_subpics_minus1; i++) {</code>	
<code>sps_subpic_ctu_top_left_x[i]</code>	<code>u(v)</code>
<code>sps_subpic_ctu_top_left_y[i]</code>	<code>u(v)</code>
<code>sps_subpic_width_minus1[i]</code>	<code>u(v)</code>
<code>sps_subpic_height_minus1[i]</code>	<code>u(v)</code>
<code>sps_master_subpic_flag[i]</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	

...	
-----	--

En una variante de esta realización, una primera disposición puede convertirse en unidades para especificar una segunda disposición. En un ejemplo, la imagen se particiona en una cuadrícula de subimágenes rectangulares y una segunda disposición utiliza las unidades de la cuadrícula de la primera disposición para especificar porciones de la imagen que tienen un estado maestro. La FIG. 7 ilustra un ejemplo donde la imagen se particiona en subimágenes 0 a 5 y las subimágenes 0 a 2 se indican como subimágenes maestras señalando el índice de la subimagen inicial y la de la subimagen final que se indican como subimágenes maestras, por lo que en este ejemplo se señalan los valores 0 y 2.

Realización 6: Señalización de un área maestra

En esta realización, la posición espacial de un área maestra se indica en la imagen independientemente de las particiones de subimagen y la(s) subimagen(es) situadas en esa área maestra serán la(s) subimagen maestra(s). Un área maestra se define como un área en la imagen donde todas las porciones que se cruzan con esta área tienen un estado maestro. Un ejemplo se ilustra en la FIG. 8, donde se indica un área maestra a la izquierda. Se asumen disposiciones de primera y segunda subimagen donde para la disposición de la primera subimagen con 4 subimágenes en este ejemplo la subimagen 0 indica una subimagen maestra, ya que la subimagen 0 está situada en el área maestra.

Para la segunda disposición de subimágenes, esta vez con 6 subimágenes, ambas subimágenes 0 y 1 están en el área maestra, por lo que ambas se indican como subimágenes maestras.

En una variante de esta realización, una subimagen se indica como subimagen maestra si se superpone total o parcialmente con la parte de la imagen indicada como área maestra, como se ilustra en la FIG. 9, donde las subimágenes 0, 1, 3 y 4 se indican subimágenes maestras ya que se solapan total o parcialmente con la parte de la imagen indicada como área maestra.

Realización 7: Hay exactamente una porción en la imagen con el estado

10 A

En esta realización hay una porción en la imagen que tiene el estado A y cuando una porción de la imagen está indicada con el estado A, ninguna otra porción de la imagen se indica con el estado A. Un ejemplo es cuando en el flujo de bits se indica que solo una subimagen tiene un estado maestro.

15 En una variante de esta realización, cuando una porción de la imagen se indica con el estado A, se infiere que todas las demás porciones de la imagen no tienen el estado A.

En una realización solo hay una subimagen en la imagen que tiene estado maestro y todas las demás subimágenes en la imagen no tienen estado maestro.

20 En un ejemplo, un valor identificador se señala en un encabezado o conjunto de parámetros para especificar cuál subimagen es la única subimagen con estado maestro.

En otra variante de esta realización hay exactamente una porción en la imagen que tiene el estado A. En una realización, un elemento sintáctico en un encabezado o conjunto de parámetros igual a un primer valor indica que hay una subimagen maestra por imagen.

- 5 En una variante de esta realización, es la primera porción de la imagen la que se indica que tiene el estado A. La primera porción puede definirse como la primera porción del flujo de bits, la primera porción en un orden de barrido progresivo o la porción con el ID más bajo. En un ejemplo, un indicador en un conjunto de parámetros especifica que la subimagen con índice de subimagen
- 10 igual a 0 es una subimagen maestra.

En un ejemplo, exactamente una porción de la imagen tiene estado A o estado maestro, y el decodificador para decodificar la imagen en este ejemplo sigue las siguientes etapas:

- (1) Decodificar un primer valor de índice de un elemento sintáctico del
- 15 primer índice en el flujo de bits;
- (2) Determinar, usando el valor del índice, la porción de la imagen con el estado igual al estado A o al estado maestro;
- (3) Decodificar la porción de la imagen con el estado igual al estado A en respuesta a que la primera porción tenga estado A o el estado maestro;
- 20 (4) No decodificar ninguna otra porción de la imagen.

En otra variante de esta realización, la imagen puede tener otras porciones, pero otras porciones de la imagen no tienen un estado A. En un

ejemplo, la única porción con estado maestro es la subimagen superior izquierda de la imagen. En otro ejemplo, la única porción con estado maestro es la subimagen con ID de subimagen igual a valor X.

En una realización, la porción de la imagen es una subimagen y la
5 existencia de solo una subimagen maestra es conocida por el decodificador por defecto o sin una señalización dedicada.

En otro ejemplo de esta realización, un elemento sintáctico en el flujo de bits igual a un primer valor indica que solo hay una porción en la imagen con estado maestro. Este elemento sintáctico puede señalizarse en un encabezado
10 o en un conjunto de parámetros.

Realización 8: Indicación de subimagen maestra en el propio encabezado

En esta realización, se señala una indicación en el encabezado de la porción actual que indica si la porción actual tiene estado A o no. En un ejemplo, la porción es un segmento y la indicación de que la porción actual tiene o no el
15 estado A se señala en el encabezado del segmento.

En una variante de esta realización, la porción que no tiene estado A solo lleva la indicación de que no tiene estado A y no hay más datos de muestra codificados. En un ejemplo de esta variante, un segmento que no es un segmento maestro solo lleva la indicación de que no es un segmento maestro en
20 el encabezado de segmento, y no hay otros datos de segmento.

En otra variante de esta realización, la indicación especifica si la porción actual de la imagen tiene estado A o estado B (en esta variante existen más de un estado).

Realización 9: Más de un nivel, jerarquía de subimagen: estado A, B, ...

- 5 En esta realización, basándose en cualquiera de las realizaciones anteriores, hay más de un estado en la imagen, por ejemplo, estado A, estado B y estado C, cada uno describiendo un estado o característica específica.

En una realización, hay más de un nivel posible de jerarquía para el estados. Por ejemplo, una porción con estado A tiene un estado superior que
10 una porción con estado B, y una porción con estados B tiene un estado superior que una porción con estados C. Los diferentes niveles de estados pueden considerarse aquí prioridades distintas.

Un ejemplo con múltiples estados incluye la forma de priorizar qué enviar/pasar si las condiciones de la red son limitadas o priorizar qué decodificar,
15 por ejemplo, si la decodificación va retrasada por alguna razón.

Otro caso de uso de múltiples estados podría ser al fusionar vídeos con diferentes tasas de fotogramas, como un collage de vídeos de vigilancia donde las cámaras tienen diferentes tasas de fotogramas. Por ejemplo, supongamos que un flujo de subimagen cam1 de 60 fps se fusiona en paralelo con un flujo de
20 subimagen cam2 de 30 fps. Cam1 es la señal principal y está marcada con el estado A en el vídeo paralelo fusionado, mientras que cada dos fotogramas de cam2 en el vídeo paralelo fusionado está marcado como estado B o estado C.

Los fotogramas con estado B tienen datos de muestra codificados y los fotogramas con estado C no tienen datos de muestra codificados, por lo que no están decodificados.

En una versión de esta realización, los múltiples estados se agrupan en representaciones (o subconjuntos de la imagen). Por ejemplo, una primera representación comprende porciones solo con estado A, una segunda representación comprende porciones con estado A o B y una tercera representación comprende porciones con estado A o C.

En un ejemplo, imagina una imagen en el vídeo con textura y profundidad empaquetadas en paralelo, cada una con un área de región de interés (ROI) como se ilustra en la FIG. 10.

Se podría entonces dividir esta imagen en porciones con cuatro estados diferentes: (1) estado A: ROI de textura (porción 1); (2) estado B: textura no ROI (porción 2); (3) estado C: ROI de profundidad (porción 3); y (4) estado D: profundidad no retorno de inversión (porción 4). Las representaciones para subconjuntos de esta imagen podrían entonces ser: representación 1: estado A (textura ROI); representación 2: estado A+B (textura completa); representación 3: estado A+C (textura ROI con profundidad); representación 4: estado C+D (profundidad completa); y representación 5: estado A+B+C+D (textura completa con profundidad).

En otra versión de esta realización, una porción puede tener múltiples estados al mismo tiempo. Esto puede verse como una alternativa a los estados

agrupados. Con la misma imagen de ejemplo que la anterior se podían usar los siguientes estados: (1) estado A: textura; (2) estado B: profundidad; (3) estado C: ROI, donde las porciones tendrían los siguientes estados: porción 1: estado A y C; porción 2: estado A; porción 3: estado B y C; y porción 4: estado B.

- 5 Las representaciones comprenden entonces los segmento que tienen las siguientes combinaciones de estados:

TABLA 11

Representación	Combinaciones de estados
1. Textura ROI	Ambos estados A y C
2. Textura completa	Al menos el estado A
3. Textura ROI con profundidad	Al menos el estado C
4. Profundidad total	Al menos el estado B
5. Imagen completa con profundidad	Cualquier combinación de estados A, B y C

- Otro ejemplo de tener múltiples estados es el ejemplo de orientación de la realización 3. En ese ejemplo se podrían definir los siguientes dos estados: estado A: alineación vertical y estado B: alineación horizontal, donde la subimagen 4 tendría el estado A y B, las subimágenes 1 y 7 solo tendrían estado A, las subimágenes 3 y 5 solo tendrían estado B, y las subimágenes 0, 2, 6 y 8 no tendrían estado.
- 10

Realización 10: Condición de habilitación:

En esta realización, la decodificación de la porción de la imagen con estado igual a estado X es condicional, por ejemplo, por otro elemento sintáctico en el flujo de bits. En una realización, se indica un estado de subimagen maestra para al menos una porción de la imagen y, si se cumplen uno o más criterios (es decir, se cumple un criterio), solo se decodifican las subimágenes de la imagen con estados de subimagen maestra, de lo contrario, todas las subimágenes se decodifican. En una de las realizaciones, se cumplen los criterios si un indicador en un conjunto de parámetros se establece a un valor específico (por ejemplo, 1), el perfil, el estrato y/o el nivel del flujo de bits satisface una condición (por ejemplo, si el flujo de bits tiene un nivel por debajo de un umbral, entonces se cumple la condición), y/o el decodificador tiene los recursos para decodificar a tiempo todas las porciones. En otro ejemplo, si se cumple o no el criterio depende de uno o más elementos sintácticos en el flujo de bits.

En un ejemplo, cuando el codificador no codifica una primera porción de la imagen, se señala un estado A para esa porción de la imagen en el flujo de bits y un indicador en un conjunto de parámetros se establece igual a un primer valor. En este ejemplo, el decodificador decodifica el indicador siendo igual a un primer valor y, en base a ese, no decodifica la primera porción de la imagen que tiene un estado A. En otras palabras, el decodificador está configurado de modo que, si determina que una porción particular tiene un estado A y el indicador es igual al primer valor, entonces el decodificador no decodifica la porción particular;

de lo contrario (es decir, el estado no es A o el indicador no es igual al primer valor) el decodificador decodifica la porción en particular.

Realización 11: Un mecanismo para conjuntos de parámetros cuando hay subimágenes maestras indicadas: poner un conjunto de parámetros en espera y recuperar otro conjunto de parámetros, cambiar entre modos 1 y 2 en cada modo usando un conjunto de parámetros diferente, acción señalizada y acción por defecto donde la acción señalizada implica usar un conjunto de parámetros y la acción predeterminada implica usar otro conjunto de parámetros en la decodificación de la imagen.

En esta realización hay un primer y un segundo conjunto de parámetros en el flujo de bits y, si hay cualquiera porciones de imagen con estado A en la imagen, entonces el primer conjunto de parámetros se utiliza en el procedimiento de decodificación de la imagen; de lo contrario (es decir, no hay porción de imagen con estado A en la imagen) se usa el segundo conjunto de parámetros en el procedimiento de decodificación de la imagen. El conjunto de parámetros que no se utiliza para decodificar la imagen se denomina un conjunto de parámetros “en espera”, que luego puede recuperarse (usarse de nuevo para el procedimiento de decodificación de una nueva imagen) basada en el estado de las porciones de la imagen que se va a decodificar.

Por ejemplo, lo siguiente puede aplicarse para poner el conjunto de parámetros A en espera y luego recuperarlo basado en tener o no una subimagen maestra en la imagen que se está decodificando:

a) El conjunto de parámetros A es válido para una imagen particular (una imagen que tiene $POC = i$) que no tiene ninguna subimagen maestra,

b) Subimagen maestra(s) se indica en otra imagen (por ejemplo, una imagen con $POC = i + 1$), el conjunto de parámetros A no es válido para la imagen
 5 $POC = i + 1$, se utilizan otros medios o mecanismos de inferencia u otro conjunto de parámetros B para la imagen $POC = i + 1$,

c) Subimagen maestra(s) no se indica en la imagen $POC = i+2$, el conjunto de parámetros A vuelve a ser válido para imagen con $POC = i+2$ sin señalización de nuevo.

10 En una realización, se señalizan dos conjuntos de parámetros de decodificación (DPS o unidad NAL de información de capacidad de decodificación (DCI)) que contienen la información de capacidad de decodificación, uno para usarse cuando hay subimágenes maestras en la imagen y otro para usarse cuando no hay subimágenes maestras en la imagen.

15 En una realización, el primer y el segundo conjunto de parámetros tienen valores identificadores o ID de conjunto de parámetros iguales.

En una realización, los estados de los segmento de imagen se señalan en el PPS, y hay dos SPS con los mismos ID de conjunto de parámetros, y uno se usa en el procedimiento de decodificación si hay al menos una porción con
 20 estado A en la imagen (los estados de las porciones se señalan en el PPS de la imagen), mientras que el otro SPS se usa en el procedimiento de decodificación de la imagen si no hay ninguna porción con el estado A en la imagen (los estados

de la porción se señalan en el PPS de la imagen). En este ejemplo, los elementos sintácticos del conjunto de parámetros superior, aquí SPS, necesarios para el análisis correcto del conjunto de parámetros inferior, PPS aquí, son los mismos en ambos SPS.

- 5 En otro ejemplo, hay porciones con estado A que el decodificador puede abstenerse de decodificar. Si el decodificador decide no decodificar esas porciones, entonces el decodificador utiliza una disposición para las porciones especificadas en un primer SPS, pero si decide decodificar esas porciones, entonces se utiliza una disposición de segunda porción especificada en un
- 10 segundo SPS. Ambos SPS están en el flujo de bits con el mismo ID.

En otro ejemplo más, un indicador en el encabezado de imagen especifica si la imagen tiene una porción con estado A o no. Si la imagen tiene al menos una porción con estado A, entonces se utiliza un primer PPS para decodificar la imagen; de lo contrario, se emplea un segundo PPS para decodificar esa imagen.

- 15 Realización 12: Una entrada indica qué subimágenes serán decodificadas.

- En esta realización, el decodificador 104 puede decodificar solo subimágenes indicadas como con estado A en el flujo de bits en respuesta a una entrada sobre el dispositivo 103. La entrada puede especificar aspectos
- 20 relacionados con la decodificación de recursos, tamaño /resolución/orientación/relación de pantalla, memoria o aspectos de procesamiento, configuración de energía o pantalla, etc. En un ejemplo, el

dispositivo 103, que aloja el decodificador 104, está funcionando en modo ahorro de energía y, como resultado de recibir entrada que indica que el dispositivo está en modo de ahorro de energía, el decodificador solo decodifica subimágenes en el flujo de bits que están marcadas como subimágenes maestras.

5 Realización 13: El decodificador garantiza la decodificación exacta a nivel de bits y la emisión solo de las porciones maestras.

En una realización, el decodificador garantiza decodificar solo las porciones de imagen que se indican como de estado A. En otra realización, la decodificación para el subconjunto de las porciones de imagen indicadas con
10 estado A está garantizada exacta a nivel de bits, lo que significa que los valores de muestra en las porciones decodificadas tienen garantizado los valores de muestra previstos por el codificador, pero la decodificación de las porciones de imagen que no tienen estado A puede no ser exacta a nivel de bits lo que significa que los valores de muestra en las porciones sin estado A pueden ser diferente
15 de los valores de muestra previstos por el codificador. En una realización, incluso si hay toda la información necesaria para decodificar una porción sin estado A en el flujo de bits, el decodificador no decodificará esa porción.

En un ejemplo, solo las subimágenes indicadas como subimágenes maestras son decodificadas por el decodificador. El codificador podría codificar
20 valores de muestra para todas las porciones o solo las porciones con estado maestro.

En una variante de esta realización, el decodificador puede no decodificar las porciones de imagen que no tengan estado A. En otra variante de esta realización, el decodificador no decodificará las porciones de imagen que no tengan estado A.

5 Realización 14: Aspectos del codificador

En esta realización, el codificador analiza diferentes porciones de la imagen e indica valores de estado, por ejemplo, define un nivel de prioridad como alta, media y baja prioridad e indica para cada porción un estado A si la porción tiene alta prioridad, un estado B si la porción tiene una prioridad media o un
10 estado C si la porción tiene baja prioridad. El decodificador entonces decodifica todas las porciones o solo un conjunto de ellas basado en sus valores de estado indicados.

En una realización, una o más porciones de la imagen son indicadas por el codificador como teniendo un valor de estado y otras porciones no tienen valor
15 de estado.

En otro ejemplo hay al menos dos valores de estado A y B y cada porción de la imagen está indicada por el codificador con estado A o estado B.

En una realización, la indicación del valor de estado de una porción de la imagen se basa en las propiedades de los valores de la muestra en esa imagen
20 o en la variación relativa de los valores de la muestra de una imagen a otra, como el movimiento de los objetos en la imagen.

En una variante de esta realización, el codificador recibe entradas que luego son usadas por el codificador para particionar la imagen en porciones y/o indicar el estado de una o más porciones de la imagen. En un ejemplo, esta entrada proviene de otro procedimiento que procesa el contenido de la imagen o de un entorno externo, como un archivo de configuración o preferencia de un usuario, como que un área particular de la imagen se especifica al codificador como una región de interés (RoI) o que tiene el contenido más interesante en comparación con otras porciones de la imagen.

Realización 15: Preprocesamiento para establecer particiones de porciones y valores de estado

En esta realización, una etapa de preprocesamiento (por ejemplo, un módulo de red neuronal (NN)) procesa el contenido y establece las particiones en las imágenes (por ejemplo, particiones de subimágenes) que particiona la imagen en porciones de imagen y/o establece el estado con relación a cada porción de imagen. Este preprocesamiento puede formar parte del codificador o realizarse en una entidad separada. Los valores de partición de imagen y estado pueden asignarse una vez durante toda la secuencia de vídeo o actualizarse durante dicha secuencia. El codificador puede absorber los valores de partición y estado de la etapa de preprocesamiento tal como están o modificarlos antes de usarlos en el procedimiento de codificación.

Realización 16: Aspectos de red

En esta realización, un nodo de red 1104, ilustrado en la FIG. 11, recibe porciones de una imagen (por ejemplo, una imagen codificada) de un emisor 1102 (por ejemplo, codificador 102) y decide qué porciones de la imagen reenviar al dispositivo 103 (etiquetado como “receptor” en la FIG. 11). La decisión se basa en el estado de cada porción. En una realización, cuando la capacidad de red es reducida, el nodo de red 104 puede decidir reenviar solo las porciones con estados alto (por ejemplo, solo aquellas porciones que tengan un valor indicador mayor o igual a un valor umbral). Por ejemplo, cuando la capacidad de red es reducida, el nodo de red 104 puede decidir reenviar solo las porciones que están indicadas como subimagen “maestra” o reenviar solo porciones que tengan un estado incluido en un conjunto determinado de estados (por ejemplo, se reenvían prioridades media y prioridad alta, pero no prioridad baja).

En el ejemplo mostrado en la FIG. 11, el emisor 1102 envía hacia el receptor 103 un flujo de bits que comprende seis subimágenes de una imagen (por ejemplo, seis unidades NAL VCL, cada una correspondiente a una diferente de las seis subimágenes). El nodo de red 1104, que está en la ruta de comunicación entre el emisor 1102 y el receptor 103, detecta que hay una capacidad de red reducida para el dispositivo 103, determina del flujo de bits que hay dos subimágenes de estado superior, por ejemplo, estado de subimagen maestra, en la imagen y determina priorizar las dos subimágenes con mayor

estado y solo pasan por las unidades NAL necesarias para decodificar estas dos subimágenes al dispositivo 103.

En una versión de esta realización, la prioridad de las unidades NAL necesarias para decodificar las subimágenes de mayor estado se ha marcado
5 además con mayor prioridad en la capa de sistemas, la capa de red y/o la capa de red de acceso radioeléctrico (RAN). En una versión, los paquetes de datos que comprenden estas unidades NAL han sido adicionalmente marcadas usando LFS (rendimiento escalable de baja latencia y baja pérdida). Cualquier nodo en RAN que sirva a un flujo de datos compatible con L4S puede establecer los bits
10 de Notificación Explícita de Congestión (ECN) en el encabezado IP para el flujo si se experimenta congestión. En otra versión, los paquetes de datos que comprenden las unidades NAL (o porciones de ellas) han sido marcados con un indicador de prioridad en una señalización conjunta de Unidad de Datos de Paquetes (PDU) en el protocolo RTP, de modo que las unidades NAL necesarias
15 para decodificar las dos subimágenes de mayor estado tienen mayor prioridad que las unidades NAL para las otras subimágenes.

Realización 17: Conjunto de porción de emisión (OPS)

En una realización, un OPS activo especifica las porciones de imagen que deben ser decodificadas. Un OPS activo también puede especificar las porciones
20 de imagen que deben ser decodificadas y emitidas.

En una realización, los OPS se señalizan para toda una secuencia de imágenes, por ejemplo, en un conjunto de parámetros como SPS, VPS o en DCI.

En otra realización, los OPS pueden ser señalizados con mayor frecuencia, por ejemplo, para una sola imagen o un conjunto de imágenes en una secuencia. En esta realización, los OPS pueden señalizarse en un encabezado, como el encabezado de imagen o encabezado de segmento, o en un conjunto de parámetros como el PPS o el APS. En una de las realizaciones, los OPS se

5 señalizan en uno o más mensajes SEI.

En una realización, todos los datos no pertenecientes a la capa de codificación de vídeo (no VCL), por ejemplo, conjuntos de parámetros, encabezado de imagen, mensajes SEI de nivel de imagen y de nivel superior, se

10 decodifican del flujo de bits, pero para los datos VCL, por ejemplo, datos codificados de segmentos, solo se decodifican los datos VCL correspondientes a una porción de imagen que se identifica por un OPS activo.

En una realización, un OPS comprende un conjunto de ID de porción. Un ejemplo se muestra en la TABLA 12 y la sintaxis de ejemplo de cómo se puede

15 señalar esto se muestra en la TABLA 13.

TABLA 12

OPS ID	ID de porción
OPS0	1-5
OPS1	1-5, 9-12
OPS2	1-12
OPS3	1-8

TABLA 13

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
}	
}	
}	

Aquí “num_opss_minus1” más 1 especifica el número de OPS señalizados en el output_portion_sets_structure(). Esta estructura puede, por ejemplo, señalizarse en un conjunto de parámetros como el SPS o el VPS.

“num_portions_minus1 [i]” más 1 especifica el número de porciones activas en el i-ésimo OPS. Las porciones activas son las porciones que deberían ser decodificadas para el OPS. “portion_id [i][j]” especifica el ID de la porción activa j-ésima y el i-ésimo OPS.

En otra realización (véase el ejemplo en la Tabla 14), un valor indicador, por ejemplo, un indicador por ejemplo portion_active_flag[i] [j], se señala en el OPS para cada porción de una imagen, en la que un primer valor del indicador,

por ejemplo 1, especifica que la porción debe ser decodificada, y un segundo valor, por ejemplo, 0, especifica que la porción no debe ser decodificada.

TABLA 14

output_portion_sets_structure() {	
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_in_picture_minus1[i]	u(v)
for(j = 0; j <= num_portions_in_picture_minus1[i]; j++) {	
portion_active_flag [i][j]	f(1)
}	
}	
}	

- 5 En una realización, cada OPS de uno o más OPS se utiliza para especificar qué porciones debe decodificar un decodificador conforme para generar un conjunto de porciones para un punto de operación dado. Cuando se activa (o selecciona un OPS), se dice que es un OPS activo, y un decodificador conforme decodificará el conjunto especificado de porciones para el punto de
- 10 operación correspondiente. En una realización, un conjunto de porciones que un decodificador debe poder decodificar para emitir una emisión válida es el mismo que el conjunto de porciones de emisión. En otra realización, un conjunto de

porciones que un decodificador debe poder decodificar puede incluir más porciones de las que se emiten, es decir, no todas las porciones decodificadas son emitidas. En esta realización, para cada porción incluida en el OPS, también se especifica si la porción es emitida o no. En una realización, se señala explícitamente para cada porción a decodificar si también debe ser emitida. Como ejemplo, véase la TABLA 15.

TABLA 15

	Porciones a decodificar	Porciones a emitir
OPS0	1-5	1-5
OPS1	1-5, 9-12	1-5, 9-12
OPS2	1-12	1-10
OPS3	1-8	1-6

Un ejemplo de sintaxis de cómo esto puede señalizarse se muestra en la TABLA 16.

TABLA 16

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)

for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
output_portion_flag [i][j]	u(1)
}	
}	
}	

La “portion_id [i][j]” como en la sintaxis anterior especifica el ID de la porción j-ésima activa en el i-ésimo OPS y la “output_portion_flag [i][j]” especifica si esa porción debe ser emitida o no, por ejemplo solo decodificada pero no emitida.

Un decodificador puede realizar todos o un subconjunto de las siguientes etapas para decodificar una imagen que comprende al menos dos porciones de imagen:

1. Obtener un flujo de bits.
2. Obtener información de al menos un conjunto de porción emitida (OPS), donde un OPS especifica un conjunto de porciones de la imagen. Un OPS puede especificar qué porciones de una imagen debe decodificar para un punto de operación dado. El OPS puede además comprender información sobre cuál de las porciones decodificadas también debe emitir desde el decodificador. Una porción puede ser una o más porciones de una porción codificada independientemente de otras porciones, una subimagen, un segmento, una

tesela, una CTU, un conjunto de teselas o un grupo de teselas. La información del OPS puede Decodificarse del flujo de bits, por ejemplo, desde un encabezado como un encabezado de segmento o un encabezado de imagen, o en un conjunto de parámetros como SPS, VPS, PPS, APS, en un DCI o en un mensaje SEI. Un OPS puede especificar un punto de operación para el decodificador.

3. Seleccionar un primer OPS de al menos uno de los OPS. La indicación de qué OPS usar puede ser señalizada en el flujo de bits o determinada por otros medios externos, por ejemplo, mediante entrada del usuario, codificada en el decodificador o especificada por especificaciones externas.

4. Determinar desde el primer OPS que una primera porción debe ser decodificada. Esto puede indicarse con la presencia de un identificador para la primera porción en el primer OPS, o puede indicarse con un indicador para la primera porción del primer OPS que tiene un primer valor, por ejemplo, 1.

5. Determinar del primer OPS que una segunda porción no debe ser decodificada. Esto puede indicarse con la falta de presencia de un identificador para la segunda porción en el primer OPS, o puede indicarse con un indicador para que la segunda porción en el primer OPS tenga un segundo valor, por ejemplo, 0.

6. En respuesta a determinar desde el primer OPS que una primera porción debe ser decodificada, decodificar la primera porción.

7. En respuesta a la determinación del primer OPS de que no debe ser decodificada una segunda porción, no decodificar la segunda porción.

Agrupaciones

En una realización, las porciones se agrupan en uno o más grupos donde los grupos se señalan en el OPS. Un grupo aquí puede verse como que las porciones de imagen en un grupo tienen un estado particular. Un ejemplo de agrupación de porciones se ilustra en la FIG. 4B, que comprende cuatro porciones numeradas del 0 al 3, donde la porción 0 y 1 pertenecen a un mismo grupo. Las porciones agrupadas 0 y 1 pueden considerarse que tienen un estado específico. Las porciones 2 y 3 no tienen el estados específico.

El estado (o pertenencia al grupo) puede ser señalado para cada porción de imagen, por ejemplo, en el encabezado de segmento o en el encabezado de unidad NAL. El estado también puede señalizarse para cada porción de imagen en una imagen en un bucle de un encabezado, por ejemplo, el encabezado de imagen, o en un conjunto de parámetros, como el VPS, SPS, PPS o APS.

En una variante, cada OPS se expresa como un estado. Es decir, al especificar un estado particular, se especifica un conjunto de porciones de imagen (es decir, las porciones de imagen que tienen dicho estado particular). En otra variante, cada conjunto de porciones de imagen se expresa como si tuviera uno o más estados.

En una realización, el decodificador decodifica una primera porción de la imagen en respuesta a que la primera porción tiene un estado específico (perteneciente a un determinado grupo) y no decodifica una segunda porción de

la imagen en respuesta a que la segunda porción no tiene el estado específico (no pertenece a ese grupo concreto).

Por ejemplo, un flujo de bits puede contener información de OPS donde un OPS está asociado a un estado o ID de grupo particular. El flujo de bits puede
5 contener además información de partición de imágenes donde las imágenes se
particionan en partes. Esta información de partición de imagen puede asociar a
cada partición con un ID de estado o de grupo. Además, la información de
partición de imágenes puede asociar cada partición con un ID de partición. El
flujo de bits además contiene imágenes codificadas donde cada imagen
10 codificada contiene porciones codificadas. Las porciones codificadas contienen
información de porción que comprende un ID de partición para cada porción.
Ahora, un decodificador que usa el único OPS al decodificar el flujo de bits
comprobaría para cada porción codificada el ID de partición y, usando el estado
o el ID de grupo, determina si decodificar o no la porción codificada. Si el ID de
15 estado o grupo coincide con el OPS, entonces el decodificador decodifica la
porción codificada.

En un ejemplo de esta realización, el grupo de porciones que se
decodifican y se emiten se basa en la orientación del dispositivo de visualización,
donde el dispositivo de visualización puede ser, por ejemplo, un teléfono o un
20 televisor. En un caso de uso de ejemplo, como se ilustra en la FIG. 4A, una
imagen consiste de 9 subimágenes (numeradas del 0 al 8 en la FIG. 4A) y la
imagen es más grande que la que se muestra en cualquier pantalla. Si se

muestran verticalmente, solo se muestran las porciones 1, 4 y 7 y, si se muestran horizontalmente, solo se muestran las porciones 3, 4 y 5. En este ejemplo, las porciones 1, 4 y 7 podrían indicarse como teniendo un primer estado específico (perteneciente a un primer grupo) para una pantalla vertical y las porciones 3, 4 y 5 podrían indicarse como teniendo un segundo estado específico (perteneciente a un segundo grupo) para una pantalla horizontal. Las porciones 0, 2, 6 y 8 nunca se muestran y en una realización no hay datos VCL para las porciones 0, 2, 6 y 8. En otra realización, la porción 4 tiene un estado específico A y las porciones 1 y 7 tienen un estado específico B y las porciones 3 y 5 tienen un estado específico C. Un decodificador puede entonces seleccionar o recibir instrucciones para decodificar y emitir la porción con estado A y bien la porción con estado B o la porción con estado C depende, por ejemplo, de la orientación de la pantalla.

Esto puede hacerse sin porciones de la imagen, pero la ventaja podría ser que las porciones que no se mostrarán no necesitan ser decodificadas. A menudo está bien esperar un segundo si la orientación de la pantalla está invertida.

En otra realización, hay más de un estado en la imagen, por ejemplo, estado A, estado B y estado C, cada uno describiendo un estado o rasgo específico, y las características pueden combinarse.

Un ejemplo con múltiples estados incluye los medios para priorizar qué enviar/transmitir si las condiciones de la red son limitadas o priorizar qué decodificar, por ejemplo, si la decodificación se retrasa por alguna razón.

Otro caso de uso de múltiples estados podría ser al fusionar vídeos con
5 diferentes tasas de fotogramas, como un collage de vídeos de vigilancia donde las cámaras tienen diferentes tasas de fotogramas. Por ejemplo, supongamos que un flujo de subimagen cam1 de 60 fps es fusionado paralelo con un flujo de subimagen cam2 de 30 fps. Cam1 es la señal principal y está marcada con el estado A en el vídeo paralelo fusionado, mientras que cada dos fotogramas de
10 cam2 en el vídeo fusionado en paralelo está marcado como estado B o estado C. Los fotogramas con estado B tienen datos de muestra codificados y los fotogramas con estado C no tienen datos de muestra codificados, por lo que no están decodificados.

En una realización, los múltiples estados se agrupan en representaciones
15 (o subconjuntos de la imagen). Por ejemplo, una primera representación comprende solo porciones con estado A, una segunda representación comprende porciones con estado A o B y una tercera representación comprende porciones con estado A o C.

En otro ejemplo, imagina una imagen en el vídeo con textura +
20 profundidad empaquetada en paralelo, cada una con un área de región de interés (ROI) (véase, por ejemplo, la FIG. 10). Se podría dividir esta imagen en porciones con cuatro estados diferentes:

Estado A: ROI de texturas (porción 1)

Estado B: Textura no ROI (porción 2)

Estado C: ROI de profundidad (porción 3)

Estado D: Profundidad no ROI (porción 4)

- 5 Las representaciones de subconjuntos de esta imagen podrían entonces señalarse en OPS:

OPS 1: estado A (textura ROI)

OPS 2: estado A+B (Textura completa)

OPS 3: estado A+C (textura ROI con profundidad)

- 10 OPS 4: estado C+D (Profundidad total)

OPS 5: estado A+B+C+D (Textura completa con profundidad)

- El conjunto de conjuntos de porción de imagen, u OPS, puede llamarse, por ejemplo, “conjunto de estado de emisión (OSS)” en esta realización. En una variante, todos los datos no pertenecientes a la capa de codificación de vídeo (no VCL), por ejemplo, conjuntos de parámetros, encabezado de imagen, nivel de imagen y mensajes SEI de nivel superior, se decodifican, pero para los datos VCL, por ejemplo, datos codificados de segmentos, solo se decodifican aquellos con un estado incluido en el OPS; en este caso, el OPS consiste de un estado.
- 15

- En otro ejemplo más, consideremos una imagen compuesta por 12 porciones, donde las porciones 1-5 tienen el estado A (es decir, pertenecen al grupo A), las porciones 6-8 tienen el estado B (es decir, pertenecen al grupo B), y las porciones 9-12 tienen el estado C (es decir, pertenecen al grupo C). En este
- 20

ejemplo, el flujo de bits define cuatro OPS con diferente combinación de estados de acuerdo con la TABLA 17 a continuación. Un ejemplo sintáctico de cómo se puede señalar esto se muestra en la TABLA 18. Los estados en el OPS especifican para qué porciones de los datos VCL deben ser decodificados. En este ejemplo, todas las porciones tienen un estado. En una variante de la realización, una porción puede tener más de un estado (es decir, una porción puede pertenecer a más de un grupo).

TABLA 17

	Estado (grupo)
OPS0	A
OPS1	A y C
OPS2	A, B y C
OPS4	A y B

10 TABLA 18

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
num_opss_minus1	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
num_groups_in_ops_minus1[i]	<code>u(v)</code>
<code>for(j = 0; j <= num_groups_in_ops_minus1[i]; j++) {</code>	

group_id [i][j]	u(v)
}	
}	
}	

Aquí “num_opss_minus1” más 1 especifica el número de OPS señalizados en la output_portion_sets_structure() “.

“num_groups_in_ops_minus1 [i]” más 1 especifica el número de grupos (con un estado específico) que hay en el i-ésimo OPS. Los grupos son los grupos de porciones que deberían Decodificarse para el OPS. “group_id [i][j]” especifica el grupo ID (estado ID) del j-ésimo grupo en el i-ésimo OPS.

En otra realización, tanto el estado como el valor indicador de la primera realización se señalan como se muestra en la Tabla 19.

10 TABLA 19

	Estado	Porciones
OPS0	A	1-5
OPS1	A y C	1-5, 9-12
OPS2	A, B y C	1-12
OPS3	A y B	1-8

En el ejemplo donde OPS utiliza estados, la emisión puede especificarse como se muestra en la Tabla 20 a continuación, y la sintaxis de cómo se puede

señalizar se muestra en la Tabla 21. Aquí OPS1 especifica que todas las porciones que tienen estado A y C se decodifican, pero solo se emiten las porciones con estado A.

TABLA 20

	Estado	Emisión
OPS0	A	Estado A
OPS1	A y C	Estado A
OPS2	A, B y C	Estado A, B, C

5

TABLA 21

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_groups_in_ops_minus1[i]	u(v)
for(j = 0; j <= num_groups_in_ops_minus1[i]; j++) {	
group_id [i][j]	u(v)
output_group_flag [i][j]	u(1)
}	
}	
}	

El “group_id [i][j]” especifica el ID de grupo del j-ésimo grupo activo (el grupo de porciones que deben decodificarse) en el i-ésimo OPS y el “output_group_flag [i][j]” especifica si las porciones de ese grupo deben ser emitidas o no, por ejemplo, solo decodificadas pero no emitidas.

- 5 En otro ejemplo, basado en el ejemplo de la FIG. 4A, se especifica un OPS para la orientación vertical y otro OPS para la orientación horizontal. Otro OPS podría incluir todas las subimágenes.

Los OPS pueden ser señalizados en el flujo de bits o transmitidos por medios externos. En una realización de la realización, los OPS se señalizan en el VPS como para los OLS. En otras realizaciones de la realización, los OPS se señalizan en otro conjunto de parámetros, como SPS o PPS, o en un encabezado, como un encabezado de imagen o un encabezado de segmento, o en un mensaje SEI.

Indicación de qué OPS aplicar

- 15 Similar al tipo de unidad NAL OPI en VVC, el flujo de bits puede contener una indicación de cuál de los OPS debe usar el decodificador (es decir, cuáles son los OPS “activos”). Esta indicación puede denominarse indicador OPS objetivo y puede decirse que indica el OPS en su alcance. Esta indicación puede, como en VVC, estar señalizada en una unidad NAL separada con su propio tipo de unidad NAL. La indicación también puede ser señalizada en un conjunto de parámetros, como VPS, SPS o PPS, en un encabezado, como un encabezado de imagen o un encabezado de segmento, o transmitirse por medios externos,
- 20

por ejemplo, establecida por entrada del usuario, ser codificada en forma fija en el decodificador, ser definida por una especificación de terceros, etc.

En una realización, se utiliza un OPS activo para determinar o especificar el punto de operación en el que debe operar un decodificador (posiblemente en combinación con otras entradas, por ejemplo, qué capas y subcapas temporales decodificar).

En una realización, se utiliza un OPS activo para determinar o especificar un punto de operación para un decodificador conforme, incluyendo qué porciones de imagen deben decodificarse y qué porciones de imagen se generarán para el punto de operación dado.

En una realización, el OPS se combina con el OLS de modo que las porciones y capas que pueden usarse para la decodificación y la emisión se especifican en la misma estructura. De manera similar, la indicación de qué OPS usar podría señalarse en la OPI junto con la indicación de qué OLS utilizar.

15 Disposición de emisión para un OPS

En esta realización se señala una disposición para el OPS que especifica la posición de las porciones decodificadas en la imagen de emisión. En una variante preferida de esta realización, la posición especificada de las porciones decodificadas en la imagen emitida es una etapa de postprocesado que no afecta a lo que se usará como imagen de referencia para decodificar otras imágenes más adelante. Un ejemplo de sintaxis de cómo se puede señalar esto se muestra en la Tabla 22 a continuación:

TABLA 22

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
portion_output_ctu_top_left_x [i][j]	u(v)
portion_output_ctu_top_left_y [i][j]	u(v)
}	
}	
}	

En el ejemplo anterior, `portion_id[i][j]` especifica qué porciones están en el i-ésimo OPS y los elementos sintácticos `portion_output_ctu_top_left_x[i][j]` y `portion_output_ctu_top_left_y[i][j]` especifican la ubicación de la esquina superior izquierda de la j-ésima porción del i-ésimo OPS en la imagen emitida.

En una variante de esta realización que utiliza la nueva disposición para posicionar las porciones decodificadas en la imagen emitida, está restringida. En un ejemplo, un `layout1` se señala para OPS1 y un `layout2` se señala para OPS2, pero el decodificador está condicionado para usar el `layout1` para las

porciones decodificadas en OPS1 solo si las porciones de OPS2 no son decodificadas por el decodificador. En este ejemplo, por ejemplo, el layout1 y el layout2 señalizados no se usarán si se decodifican porciones tanto en OPS1 como en OPS2.

- 5 En otra variante de esta realización, la señalización de las ubicaciones para las porciones de un OPS está condicionada solo a aquellas porciones que se están emitiendo. En un ejemplo mostrado en la tabla siguiente, la nueva ubicación de una porción en un OPS solo se señala si la porción está siendo emitida. En este ejemplo, si el valor del elemento sintáctico `output_portion_flag [ i ][ j ]` es igual a 1, entonces la porción que va a ser emitida y así la ubicación de esta porción en la imagen a ser emitida se señala con `portion_output_ctu_top_left_x[ i ][ j ]` y `portion_output_ctu_top_left_y[ i ][ j ]`, y si el elemento sintáctico `output_portion_flag [ i ][ j ]` es igual a 0, entonces `portion_output_ctu_top_left_x[ i ][ j ]` y `portion_output_ctu_top_left_y[ i ][ j ]` no se
- 10
- 15 señalarán.

Tabla 23

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>num_portions_minus1[i]</code>	<code>u(v)</code>

for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [l][j]	u(v)
output_portion_flag [l][j]	u(1)
si output_portion_flag [i][j] {	
portion_output_ctu_top_left_x [l][j]	u(v)
portion_output_ctu_top_left_y [l][j]	u(v)
}	
}	
}	
}	

En otra realización, un OPS, por ejemplo, OPS2 es un subconjunto de otro OPS, por ejemplo, OPS1. En esta realización, el OPS2 puede estar limitado a contener solo porciones que también están contenidas en OPS1 (está bien no tener todas) y en las que las porciones en OPS2 tienen la misma disposición que las porciones correspondientes de OPS1 en la imagen de emisión. En un ejemplo, mostrado en la tabla siguiente, la disposición de las porciones solo se 5
señaliza para el OPS del superconjunto (OPS1 en el ejemplo anterior) pero se aplica tanto al superconjunto como al subconjunto de los OPS (OPS1 y OPS2 en 10
el ejemplo anterior).

TABLA 24

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
subset_of_another_ops[i]	u(1)
Si subset_of_another_ops[i]	
superset_ops_id[i]	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
Si (!subset_of_another_ops[i]){	
portion_output_ctu_top_left_x [i][j]	u(v)
portion_output_ctu_top_left_y[i][j]	u(v)
}	
}	
}	
}	

En el ejemplo anterior, subset_of_another_ops[i] igual a 1 especifica que el (i+1)-ésimo OPS es un subconjunto de otro OPS especificado por superset_ops_id[i], y no se señala una nueva disposición para el (i+1)-ésimo

OPS, mientras que `subset_of_another_ops[ i ]` igual a 0 especifica que el (i+1)-ésimo OPS no es un subconjunto de otro OPS y se señala una nueva disposición para el (i+1)-ésimo OPS usando elementos sintácticos `portion_output_ctu_top_left_x[ i ][ j ]` y `portion_output_ctu_top_left_y[ i ][ j ]`•

5 Porciones de escalado especificadas por un OPS

En esta realización, también se señala en el flujo de bits cómo los segmento especificadas en un OPS se escalan tras la decodificación por el decodificador para ser producidas como imagen de emisión.

Un ejemplo en la siguiente tabla muestra cómo esto puede ser señalizado.

10 TABLA 25

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>num_portions_minus1[i]</code>	<code>u(v)</code>
<code>for(j = 0; j <= num_portions_minus1[i]; j++) {</code>	
<code>portion_id [i][j]</code>	<code>u(v)</code>
<code>portion_output_ctu_top_left_x[i][j]</code>	<code>u(v)</code>
<code>portion_output_ctu_top_left_y[i][j]</code>	<code>u(v)</code>
<code>portion_output_scale_x[i][j]</code>	<code>u(v)</code>
<code>portion_output_scale_y[i][j]</code>	<code>u(v)</code>

}	
}	
}	

En la tabla anterior, los elementos sintácticos `portion_output_scale_x[ i ][ j ]` y `portion_output_scale_y[ i ][ j ]` especifican el factor de escalamiento correspondiente a las dimensiones x e y de la j-ésima porción en el i-ésimo OPS.

- 5 En una variante de esta realización, la señalización de los factores de escala está condicionada. En un ejemplo, el factor de escala de una porción solo se condiciona si la porción va a ser decodificada y emitida.

Cada porción pertenece exactamente a un OPS

- 10 En esta realización, cada porción pertenece exactamente a un OPS. En un ejemplo, la división de las porciones en OPS se señala en un bucle sobre el número de porciones que indican un id de OPS para cada porción, como se muestra a continuación:

TABLA 26 (lista de los id de OPS)

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>num_portions_minus1</code>	<code>u(8)</code>
<code>para (j = 0; j <= num_portions_minus1; j++) {</code>	
<code>ops_id[j]</code>	<code>u(v)</code>

}	
...	
}	

`ops_id[ j ]` es un identificador OPS para la partición j-ésima.

Valores de muestra para porciones que no están en un OPS activo

Esta realización puede combinarse con cualquiera de las realizaciones
5 anteriores y, además, para un OPS, se señala qué valor de muestra emitir para las porciones que no están decodificadas (las porciones que no están en el conjunto de porciones activas del OPS).

En un ejemplo, una porción P de la imagen no se decodifica en el punto de operación de OPS1, y se señala un elemento sintácticos de un byte `filler_val`
10 que especifica el valor de muestra para rellenar las porciones no decodificadas, cuando la imagen de emisión incluye porciones no decodificadas como P. En este ejemplo, `filler_val = 0` especifica que los valores de muestra correspondientes a la porción P en la imagen de emisión están rellenos con blanco, y `filler_val = 255` especifica que los valores de muestra correspondientes
15 a la porción P no decodificada en la imagen de emisión están rellenos de negro y que todos los demás valores de `filler_val` especifican una escala de grises. Este ejemplo se muestra en la Tabla 27.

TABLA 27

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
filler_val	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id[i][j]	u(v)
}	
}	
}	

En el ejemplo anterior, un elemento de sintaxis `filler_val` se especifica una vez para todos los OPS.

- 5 En una variante de esta realización, la especificación de los valores de muestra de emisión para las porciones que no están decodificadas es de acuerdo con el OPS. A continuación se muestra un ejemplo donde `filler_val[ i ]` especifica el valor que se usará como relleno de los segmento no decodificados cuando se tiene el i-ésimo OPS como punto activo de operación:

TABLA 28

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
filler_val[i]	u(8)
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id[i] [j]	u(v)
}	
}	
}	

En otra variante de esta realización, los valores de muestra no decodificados en la imagen de emisión se copian de otra imagen o de porciones decodificadas de la imagen actual. En un ejemplo de esta variante mostrado en la tabla siguiente, un elemento sintáctico `copy_interval = X` especifica que para todos los puntos de operación activos definidos `OPS[ i ]` los valores de muestra no decodificados en la imagen de emisión se copian de los valores de muestra colocados de la imagen de emisión anterior (X+1)-ésimo en el orden de emisión, donde `copy_interval = 0` especifica copiar de la imagen de emisión anterior.

TABLA 29

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
copy_interval	u(v)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id[i] [j]	u(v)
}	
}	
}	

OPS como SEI de postprocesamiento

En esta realización, OPS es una herramienta puramente de postprocesamiento, por lo que no afecta la decodificación de las imágenes, sino que señala los procedimientos que deben aplicarse a la imagen decodificada para crear imágenes de emisión. En un ejemplo de esta realización, la información del OPS se señala en un mensaje SEI e incluye la información (como la información de la distribución) de qué porciones de la imagen decodificada deben juntarse para crear una imagen de emisión. En un ejemplo de esta realización, la información OPS incluye información POC que especifica

porciones de cuales imágenes decodificadas se usarán para crear una imagen de emisión. La siguiente tabla muestra un ejemplo de un mensaje OPS SEI:

TABLA 30

output_portion_set(Tamaño de carga) {	Descriptor
...	
ops_id	UE(v)
ops_num_portions_minus1	u(v)
for(i = 0; i <= num_portions_minus1; i++) {	
ops_portion_id [i]	u(v)
ops_portion_output_ctu_top_left_x[i]	u(v)
ops_portion_output_ctu_top_left_y[i]	u(v)
}	
...	
}	

- 5 La FIG. 12 es un diagrama de flujo que ilustra un procedimiento 1200 para decodificar una imagen de un flujo de bits, la imagen comprende una primera porción y una segunda porción. El procedimiento 1200 puede comenzar en la etapa s1202. La etapa s1202 comprende obtener, del flujo de bits, la primera información de estado (por ejemplo, un primer valor indicador) correspondiente
- 10 a la primera porción de la imagen, en la que la primera información de estado no corresponde a la segunda porción. La etapa s1204 comprende determinar si se

decodifica o no la primera porción de la imagen basándose en la primera información de estado (por ejemplo, decodificar la primera porción de la imagen en respuesta a determinar que la primera información de estado indica que la primera porción tiene un primer estado y abstenerse de decodificar la primera porción de la imagen en respuesta a determinar que la primera información de estado indica que la primera porción no tiene el primer estados y se cumplen cero o más otros criterios).

La FIG. 13 es un diagrama de flujo que ilustra un procedimiento 1300 para emitir un flujo de bits que comprende una imagen (es decir, una imagen codificada), la imagen comprende una primera porción y una segunda porción. El procedimiento 1300 puede comenzar en la etapa s1302. La etapa s1302 comprende determinar si la primera porción debe ser decodificada por un decodificador. La etapa s1304 comprende, como resultado de determinar que la primera porción debe ser decodificada por un decodificador, producir información de estado codificada para la primera porción e incluir la información de estado codificada en el flujo de bits, en la que la información de estado codificada para la primera porción no corresponde a la segunda porción, y la información de estado codificada sirve para informar al decodificador de que este no debe abstenerse de decodificar la primera porción de la imagen.

La FIG. 14 es un diagrama de flujo que ilustra un procedimiento 1400 realizado por un nodo de red (por ejemplo, nodo de red 1104). El procedimiento 1400 puede comenzar en la etapa s1402. La etapa s1402 comprende recibir i)

una primera unidad (por ejemplo, unidad NAL VCL) que comprende valores de muestra codificada de imagen correspondientes a una primera porción de una imagen e ii) la primera información de estado (por ejemplo, información codificada del primer estado) correspondiente a la primera porción de la imagen, en la que la primera información de estado no corresponde a una segunda porción de la imagen. La etapa a1403 (opcional) comprende recibir una segunda unidad (por ejemplo, unidad NAL VCL) que comprende valores de muestra de imagen codificada correspondientes a la segunda porción de la imagen. La etapa s1404 comprende, basándose en la primera información de estado, decidir si se reenvía o no la primera unidad a un dispositivo receptor. En una realización, si la primera información de estado indica un primer estado de la primera unidad, entonces la primera unidad no se reenviará. Como la primera información de estado no corresponde a la segunda porción de la imagen, la decisión de si reenviar o no la segunda unidad correspondiente a la segunda porción puede ser independiente de la primera información de estado. En una realización, la primera información de estado se incluye en la primera unidad.

La FIG. 16 es un diagrama de bloques de un aparato 1600 para implementar el codificador 102, el decodificador 104 y/o el nodo de red 1104, de acuerdo con algunas realizaciones. Cuando el aparato 1600 implementa el codificador 102, el aparato 1600 puede denominarse aparato codificador; cuando el aparato 1600 implementa el decodificador 104, el aparato 1600 puede denominarse aparato decodificador; y cuando el aparato 1600 implementa el

nodo de red 1104, el aparato 1600 puede denominarse nodo de red. Como se muestra en la FIG. 16, el aparato 1600 puede comprender: circuitería de procesamiento (PC) 1602, que puede incluir uno o más procesadores (P) 1655 (por ejemplo, uno o más microprocesadores de propósito general y/o uno o más procesadores más, como un circuito integrado específico de aplicación (ASIC), matrices de puertas programables en campo (FPGA), y similares), cuyos procesadores pueden estar ubicados en una sola carcasa o en un único centro de datos o pueden estar distribuidos geográficamente (es decir, el aparato codificador 1600 puede ser un aparato de computación distribuida); al menos una interfaz de red 1648 (por ejemplo, una interfaz física o interfaz de aire) que comprende un transmisor (Tx) 1645 y un receptor (Re) 1647 para permitir que el aparato 1600 transmita datos y reciba datos de otros nodos conectados a una red 110 (por ejemplo, una red de protocolo de Internet (IP)) a la que la interfaz de red 1648 está conectada (física o inalámbricamente) (por ejemplo, la interfaz de red 1648 puede acoplarse a una disposición de antenas que comprende una o más antenas para permitir que el aparato codificador 1600 transmita/reciba datos de forma inalámbrica); y una unidad de almacenamiento (también conocida como “sistema de almacenamiento de datos”) 1608, que puede incluir uno o más dispositivos de almacenamiento no volátil y/o uno o más dispositivos de almacenamiento volátiles. En realizaciones donde PC 1602 incluye un procesador programable, puede proporcionarse un medio de almacenamiento legible por ordenador (CRSM) 1642. El CRSM 1642 puede almacenar un

programa informático (CP) 1643 que incluya instrucciones legibles por ordenador (CRI) 1644. CRSM 1642 puede ser un medio legible por ordenador no transitorio, como medios magnéticos (por ejemplo, un disco duro), medios ópticos, dispositivos de memoria (por ejemplo, memoria de acceso aleatorio, memoria flash) y similares. En algunas realizaciones, el CRI 1644 del programa informático 1643 está configurado de modo que, cuando se ejecuta en PC 1602, el CRI hace que el aparato codificador 1600 realice las etapas descritas aquí (por ejemplo, etapas descritas aquí con referencia a los diagramas de flujo). En otras realizaciones, el aparato codificador 1600 puede configurarse para realizar las etapas descritas aquí sin necesidad de código. Es decir, por ejemplo, PC 1602 puede consistir únicamente en uno o más ASIC. Por lo tanto, las características de las realizaciones descritas aquí pueden implementarse en hardware y/o software.

Resumen de diversas realizaciones

15 A1. Un procedimiento 1200 para decodificar una imagen de un flujo de bits, la imagen que comprende una primera porción y una segunda porción, el procedimiento comprende: obtener, del flujo de bits, la primera información de estado (por ejemplo, un primer valor indicador) correspondiente a la primera porción de la imagen, en la que la primera información de estado no corresponde
20 a la segunda porción; y determinar si decodificar o no la primera porción de la imagen basándose en la primera información de estado.

A2. El procedimiento de realización A1, en el que el procedimiento comprende además en obtener segunda información de estado correspondiente a la segunda porción y no correspondiente a la primera porción; y determinar si decodificar o no la segunda porción de la imagen basándose en la segunda
5 información de estado.

A3. El procedimiento de realización A1 o A2, en el que la primera porción de la imagen es una subimagen, un segmento, una tesela, una CTU, un conjunto de teselas o un grupo de teselas.

A4. El procedimiento de cualquier realización anterior, en el que la primera
10 porción es independiente de otras porciones de la imagen (por ejemplo, es independiente de la segunda porción).

A5. El procedimiento de cualquier realización anterior, en el que la primera porción comprende un conjunto de uno o más valores de muestra codificados, y el procedimiento comprende además, en respuesta a determinar que la primera
15 información de estado indica que la primera porción tiene un primer estado, decodificar el conjunto de valores de muestra codificados para producir un conjunto de valores de muestra decodificados.

A6. El procedimiento de realización A5, en el que el procedimiento comprende además en utilizar el conjunto de valores de muestra decodificados
20 para generar un conjunto de valores de muestra reconstruidos correspondientes.

A7. El procedimiento de cualquier realización anterior, en el que la primera información de estado es un valor indicador de estado, y el procedimiento

comprende además: determinar un primer estado para la primera porción basada en el valor del indicador de estado; y decodificar la primera porción en respuesta a que el primer estado es igual a un estado A o abstenerse de decodificar la primera porción en respuesta a que el primer estado no es igual al estado A.

5 A8. El procedimiento de cualquiera de las realizaciones anteriores, en el que la primera información de estado indica que la primera porción es la porción maestra (por ejemplo, la información de estado indica que ninguna otra porción de la imagen tiene un estado superior (importancia, prioridad, etc.) que la primera porción).

10 A9. El procedimiento de cualquier realización anterior, en el que la primera información de estado se decodifica de un conjunto de parámetros, un encabezado o un mensaje SEI.

A10. El procedimiento de cualquiera de las realizaciones anteriores, en el que la primera porción comprende dos o más CTU.

15 A11. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende además: obtener, del flujo de bits, la segunda información de estado (por ejemplo, un segundo valor indicador de estado) correspondiente a la segunda porción de la imagen (el segundo valor indicador de estado puede ser igual o no igual al primer valor indicador de estado); y
20 determinar si decodificar o no la segunda porción de la imagen basándose en la segunda información de estado.

A12. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende determinar no decodificar la primera porción, y el procedimiento, además, comprende generar valores de muestra reconstruidos correspondientes a la primera porción usando: valores de muestra constantes, valores de muestra por defecto, valores de muestra previamente reconstruidos, valores de muestra colocados en una o más imágenes previamente decodificadas, otros valores de muestra en una o más imágenes previamente decodificadas, otros valores de muestra en la imagen, y/o imagen predeterminada o parte de una imagen.

10 A12b. El procedimiento de la reivindicación A12, en el que el flujo de bits comprende información para indicar que los valores de muestra reconstruidos correspondientes a la primera porción deben generarse usando: valores constantes de muestra, valores de muestra por defecto, valores de muestra previamente reconstruidos, valores de muestra colocados en una o más imágenes previamente decodificadas, otros valores de muestra en una o más imágenes previamente decodificadas, otros valores de muestra en la imagen, valores de muestra interpolados, valores extrapolados y/o una imagen o parte predeterminada de una imagen.

20 A13. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende determinar no decodificar la primera porción, y además comprende generar valores de muestra reconstruidos correspondientes

a la primera porción utilizando valores de muestra generados, copiados, interpolados y/o extrapolados.

5 A14. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende determinar no decodificar la primera porción, y el procedimiento además comprende: reorganizar al menos una segunda porción de la imagen (por ejemplo, como subimagen, segmento, tesela en la imagen decodificada); reescalar al menos la segunda porción de la imagen (escala la segunda porción para cubrir un área correspondiente a la primera porción); y/o cambiar la resolución de emisión de la imagen decodificada.

10 A15. El procedimiento de cualquier realización anterior, en el que la primera información de estado se decodifica del flujo de bits en respuesta a que un tercer indicador que sea decodificado del flujo de bits y el valor del tercer indicador sea igual a un tercer valor (por ejemplo, un tercer valor que indica que el primer valor está codificado en el flujo de bits).

15 A16. El procedimiento de realización A15, en el que el tercer indicador se decodifica de un conjunto de parámetros (por ejemplo, SPS o PPS) o de un encabezado (por ejemplo, encabezado de segmento o encabezado de imagen).

20 A17. El procedimiento de cualquier realización anterior, en el que la primera información de estado indica que la primera porción tiene un estado particular (por ejemplo, estado maestro), y el procedimiento comprende además, como resultado de determinar que la primera información de estado indica que

la primera porción tiene un estado particular, determinar que ninguna otra porción de la imagen tiene dicho estado particular.

A18. El procedimiento de cualquier realización anterior, en el que obtener la primera información de estado del flujo de bits comprende decodificar la primera información de estado de un primer elemento sintáctico de índice en el
5 flujo de bits.

A19. El procedimiento de realización A18, en el que la primera información de estado indica que la primera porción tiene un estado particular (por ejemplo, estado maestro), y el procedimiento comprende además: decodificar la primera
10 porción de la imagen en respuesta a determinar que la primera porción tiene dicho estado particular; y abstenerse de decodificar todas las demás porciones de la imagen.

A20. El procedimiento de cualquier realización anterior, en el que la primera porción es un segmento de la imagen y la indicación de estado se señala
15 en un encabezado de segmento para el segmento.

A21. El procedimiento de realización A20, en el que el encabezado de segmento está contenido en una unidad NAL VCL, y la unidad NAL VCL no contiene valores de muestra codificados (es decir, no contiene datos que representen valores de muestra de imagen).

20 A22. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende además en obtener, del flujo de bits, la segunda información de estado correspondiente a la segunda porción de la imagen, la

primera información de estado indica que la primera porción tiene un primer estado particular, estado A (por ejemplo, estado maestro), la segunda información de estado indica que la segunda porción tiene un segundo estado particular, el estado B (por ejemplo, el estado no maestro), y el estado A tiene una prioridad superior al estado B.

A23. El procedimiento de realización A22, en el que el estado A tiene una prioridad superior al estado B, indica que la primera porción con estado A no se decodifica solo si la segunda porción con estado B no lo es.

A24. El procedimiento de cualquier realización anterior, en el que el procedimiento comprende además decodificar, del flujo de bits, un valor de habilitación, la determinación de si decodificar o no la primera porción se basa no solo en la primera información de estado, sino también en el valor de la habilitación.

A25. El procedimiento de realización A24, en el que si la primera información de estado tiene un primer valor particular, entonces la primera porción solo se decodifica si el valor de habilitación satisface una condición.

A26. El procedimiento de realizaciones A24 y A25, en el que el valor de habilitación especifica que solo deben decodificarse las porciones de imagen que tengan un estado particular.

A27. El procedimiento de cualquier realización anterior, en el que el flujo de bits comprende un conjunto de conjuntos de parámetros, el conjunto de conjuntos de parámetros que comprende un primer conjunto de parámetros y un

segundo conjunto de parámetros, el procedimiento comprende además seleccionar, basándose en la primera información de estado, un conjunto de parámetros del conjunto de los conjuntos de parámetros, y el procedimiento además comprende utilizar el parámetro seleccionado en un procedimiento de
5 decodificación para decodificar la imagen.

A28. El procedimiento de realización A27, en el que el flujo de bits comprende un conjunto de parámetros de imagen (PPS) para la imagen, el primer conjunto de parámetros es un primer conjunto de parámetros de secuencia (SPS) que tiene un primer identificador (ID) de SPS, el segundo
10 conjunto de parámetros es un segundo SPS que tiene el primer ID SPS, y obtener la primera información de estado del flujo de bits comprende decodificar la primera información de estado de un elemento sintáctico en el PPS.

A29. El procedimiento de realización A28, en el que el procedimiento comprende seleccionar el primer SPS como resultado de determinar que la
15 primera información de estado indica que la primera porción tiene un estado particular o seleccionar el primer SPS como resultado de determinar que ninguna porción de la imagen tiene el estado particular.

A30. El procedimiento de cualquier realización anterior, en el que el procedimiento lo realiza un decodificador de un dispositivo, y la determinación
20 de si decodificar o no la primera porción se basa no solo en la primera información de estado, sino también en la información relativa al dispositivo.

A31. El procedimiento de realización A30, en el que la información relativa al dispositivo comprende: información sobre tamaño de pantalla, información sobre resolución de pantalla, información sobre orientación de pantalla, información sobre la relación de pantalla, información sobre carga de memoria, información sobre carga de procesamiento, información sobre nivel de batería y/o información sobre consumo de energía.

A32. El procedimiento de cualquier realización anterior, en el que solo las porciones de imagen con un estado particular tienen garantizado una decodificación exacta a nivel de bits.

10 A33. El procedimiento de cualquier realización anterior, en el que la primera información de estado indica un conjunto de porción de emisión, OPS, al que pertenece la primera porción.

B1. Un procedimiento 1300 (véase la FIG. 13) para producir un flujo de bits que comprende una imagen (es decir, una imagen codificada), cuya imagen
15 comprende una primera y una segunda porción, el procedimiento que comprende: determinar si la primera porción debe ser decodificada por un decodificador; y como resultado de determinar que la primera porción debe ser decodificada por un decodificador, producir información de estado codificada para la primera porción e incluir la información de estado codificada en el flujo de
20 bits, en el que la información de estado codificada para la primera porción no corresponde a la segunda porción de la imagen, y la información de estado

codificada sirve para informar al decodificador de que el decodificador no debe abstenerse de decodificar la primera porción de la imagen.

B2. El procedimiento de realización B1, en el que determina si la primera porción debe ser decodificada por un decodificador, comprende recibir entrada
5 de un preprocesador, en la que la entrada se utiliza para determinar si la primera porción debe ser decodificada por un decodificador.

B3. El procedimiento de realización B2, en el que la entrada comprende la información de partición de imagen que identifica las porciones de la imagen y, para cada porción identificada, un estado para la porción.

10 B4. El procedimiento de realización B3, en el que la información de partición de imagen identifica una región de interés (RoI) dentro de la imagen, y el estado del RoI es un primer estado (por ejemplo, estado maestro).

C1. Un procedimiento 1400 (véase la FIG. 14) realizado por un nodo de red, el procedimiento comprende: recibir una primera unidad (por ejemplo,
15 unidad NAL VCL) que comprende valores de muestra de imagen codificada correspondientes a una primera porción de una imagen y primera información de estado (por ejemplo, primera información de estado codificada) correspondiente a la primera porción de la imagen; y, basándose en la primera información de estado, decidir si reenviar o no la primera unidad a un dispositivo receptor.

20 C2. El procedimiento de realización C1, en el que la primera información de estado comprende un primer valor indicador de estado y decidir si se reenvía o no la primera unidad a un dispositivo receptor basándose en la información de

estado, comprende determinar si el primer valor indicador de estado cumple un criterio (por ejemplo, determinar si el primer valor indicador de estado es menor que un umbral o superior a un umbral)

C3. El procedimiento de realización C2, en el que el procedimiento
5 comprende además: recibir segunda información de estado correspondiente a la segunda porción de la imagen; y, basándose en la segunda información de estado, decidir si se reenvía o no la segunda unidad al dispositivo receptor, en el que la segunda información de estado comprende un segundo valor indicador de estado que es diferente al primer valor de indicador de estado, y decidir si se
10 reenvía o no la segunda unidad a un dispositivo receptor basándose en la segunda información de estado comprende determinar si el segundo valor indicador de estado cumple dicho criterio.

D1. Un programa informático que comprende instrucciones que, al ejecutarse mediante la circuitería de procesamiento de un aparato, hace que el
15 aparato realice el procedimiento de una cualquiera de las realizaciones anteriores.

D2. Una portadora que contiene el programa informático de la realización D1, en la que la portadora es una de una señal electrónica, una señal óptica, una señal de radio y un medio de almacenamiento legible por ordenador.

20 E1. Un aparato configurado para realizar cualquiera de los procedimientos de cualquiera de las realizaciones de procedimientos anteriores.

Aunque la terminología de esta divulgación se describe en términos de VVC, las realizaciones de esta divulgación también se aplican a cualquier códec existente o futuro, que pueda usar una terminología diferente, pero equivalente.

Además, aunque aquí se describen varias realizaciones, debe entenderse
5 que se han presentado solo como ejemplo, y no como limitación. Por lo tanto, el espíritu y alcance de esta divulgación no deben limitarse por ninguna realización ejemplares descritas anteriormente. Además, cualquier combinación de los elementos descritos anteriormente en todas las posibles variaciones de estos está incluida en la divulgación, salvo que aquí se indique lo contrario o sea
10 claramente contradicha por el contexto. Como se usa aquí, “a” significa “al menos uno” o “uno o más”.

Además, aunque los procedimientos descritos arriba e ilustrados en los dibujos se muestran como una secuencia de etapas, esto se hizo únicamente con fines ilustrativos. En consecuencia, se contempla que se pueden añadir
15 algunas etapas, omitir algunas etapas, reorganizar el orden de las etapas y realizar algunas etapas en paralelo.