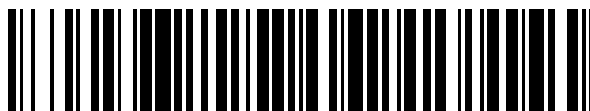


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 748 463**

51 Int. Cl.:

H04N 19/70	(2014.01)
H04N 19/61	(2014.01)
H04N 19/187	(2014.01)
H04N 19/31	(2014.01)
H04N 19/46	(2014.01)
H04N 19/30	(2014.01)
H04N 19/169	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **16.04.2013 PCT/KR2013/003204**
- 87 Fecha y número de publicación internacional: **24.10.2013 WO13157826**
- 96 Fecha de presentación y número de la solicitud europea: **16.04.2013 E 13777676 (1)**
- 97 Fecha y número de publicación de la concesión europea: **14.08.2019 EP 2840788**

54 Título: **Aparato de decodificación de vídeo**

30 Prioridad:

16.04.2012 KR 20120038870
21.06.2012 KR 20120066606
25.06.2012 KR 20120067925
02.07.2012 KR 20120071933
16.07.2012 KR 20120077012
28.09.2012 KR 20120108925
10.10.2012 KR 20120112598

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
16.03.2020

73 Titular/es:

**ELECTRONICS AND TELECOMMUNICATIONS
RESEARCH INSTITUTE (100.0%)
161 Gajeong-dong, Yuseong-gu
Daejeon-si 305-700 , KR**

72 Inventor/es:

**KANG, JUNG WON;
LEE, HA HYUN;
CHOI, JIN SOO y
KIM, JIN WOONG**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 748 463 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Aparato de decodificación de vídeo

Campo técnico

5 La presente invención se refiere al procesamiento de codificación y decodificación de vídeo y, más en particular, a un método y un aparato para decodificar información de vídeo dentro de un tren de bits.

Antecedentes de la técnica

10 R. SJOBERG ET AL, "Overview of HEVC high-level syntax and reference picture management", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 1 de enero de 2012 (2012-01-01), páginas 1-1, XP055045360, ISSN: 1051-8215, doi: 10.1109/TCSVT.2012.2223052 divulga la proporción creciente de tráfico de vídeo en las redes de telecomunicación, poniendo énfasis en la tecnología de compresión de vídeo eficaz. La codificación de vídeo de alta eficiencia (HEVC, por sus siglas en inglés) es el próximo estándar de codificación de vídeo, que proporciona considerables reducciones de la velocidad de transmisión binaria en comparación con sus predecesores. En el proceso de estandarización de la HEVC, las tecnologías tales como la subdivisión de imagen, la gestión de imagen de referencia y los conjuntos de parámetros se clasifican como "sintaxis de alto nivel". El diseño de la sintaxis de alto nivel afecta a la interfaz para sistemas y la elasticidad a los errores y proporciona nuevas funcionalidades. Este documento presenta una visión de conjunto de la sintaxis de alto nivel de la HEVC, incluyendo encabezamientos de unidad de capa de abstracción de red, conjuntos de parámetros, esquemas de subdivisión de imagen, gestión de imagen de referencia, y mensajes de información de mejora suplementaria.

15 El documento KR 2008 0007086 A divulga un método y un aparato para decodificar/codificar señales de vídeo proporcionado para realizar la codificación más eficazmente cuando se codifican por separado una imagen ancla y una imagen no ancla según una información de identificación de imagen ancla porque las dependencias intervista entre la imagen ancla y la imagen no ancla son diferentes entre sí. Un método de decodificación de señal de vídeo comprende las siguientes etapas: obtener información de etiqueta que indica si una señal de vídeo es un tren de bits codificado de vídeo multivista; si es así, obtener información de identificación que indica si una imagen codificada de una NAL (capa de abstracción de red) actual es una imagen ancla o no; y decodificar información sobre un vídeo multivista según la información de identificación.

20 Según se extiende y se ofrece a escala nacional y a escala mundial la radiodifusión de alta definición (HD, por sus siglas en inglés), muchos usuarios se están acostumbrando al vídeo de alta resolución y alta SNR (relación señal-ruido). Por consiguiente, muchas organizaciones han realizado numerosos intentos de desarrollar los dispositivos de vídeo de próxima generación. Además, dado que hay un interés creciente en la ultra alta definición (UHD, por sus siglas en inglés) con una resolución 4 veces mayor que la HDTV junto con la HDTV, existe la necesidad de una tecnología en la que se comprima y se procese un vídeo con una mayor resolución y una mayor SNR.

25 Para comprimir un vídeo, pueden utilizarse una tecnología de interpredicción, en la que un valor de un píxel incluido en una imagen actual se predice a partir de imágenes anteriores y/o posteriores en el tiempo, una tecnología de intrapredicción, en la que un valor de un píxel incluido en una imagen actual se predice utilizando información sobre un píxel incluido en la imagen actual, una tecnología de codificación entrópica, en la que se asigna un código corto a un símbolo que tiene una alta frecuencia de aparición y se asigna un código largo a un símbolo que tiene una baja frecuencia de aparición, etc.

30 La tecnología de compresión de vídeo incluye tecnología en la que se proporciona un ancho de banda de red específico en el entorno de funcionamiento limitado del *hardware* sin tener en cuenta un entorno de red flexible. Sin embargo, para comprimir datos de vídeo aplicados a un entorno de red que incluya un ancho de banda que varíe con frecuencia, es necesaria una nueva tecnología de compresión. Con este fin puede utilizarse un método de codificación/decodificación de vídeo escalable.

Divulgación

45 Problema técnico

Un objetivo de la presente invención es proporcionar un método y un aparato para describir información de extracción y escalabilidad dentro de trenes de bits en capas.

Otro objetivo de la presente invención es proporcionar un método y un aparato para representar información de escalabilidad sobre el uso de diversos trenes de bits de una manera flexible.

50 Otro objetivo más de la presente invención es proporcionar un método y un aparato para proporcionar información de extracción y escalabilidad dentro de un tren de bits en capas, de manera que la información de extracción y escalabilidad pueda transformarse de forma adaptativa en un nivel de paquete.

Solución técnica

La invención se expone en la reivindicación independiente adjunta; los ejemplos adicionales denominados realizaciones en la descripción son ejemplos ilustrativos.

Efectos ventajosos

5 De acuerdo con una realización de la presente invención, pueden proporcionarse un método y un aparato para describir información de extracción y escalabilidad dentro de trenes de bits en capas.

De acuerdo con una realización de la presente invención, pueden proporcionarse un método y un aparato para representar información de escalabilidad sobre el uso de diversos trenes de bits de una manera flexible.

10 De acuerdo con otra realización de la presente invención, pueden proporcionarse un método y un aparato para proporcionar información de extracción y escalabilidad dentro de trenes de bits en capas, de manera que la información de extracción y escalabilidad pueda transformarse de forma adaptativa en un nivel de paquete.

Descripción de los dibujos

La Figura 1 es un diagrama de bloques que muestra un ejemplo de una estructura de un aparato de codificación de vídeo según una realización ejemplar;

15 la Figura 2 es un diagrama de bloques que muestra un ejemplo de una estructura de un aparato de decodificación de vídeo según una realización ejemplar;

la Figura 3 es un diagrama conceptual que muestra esquemáticamente una realización ejemplar de una estructura de codificación de vídeo escalable (SVC, por sus siglas en inglés) que utiliza múltiples capas a la que puede aplicarse la presente invención;

20 la Figura 4 es un diagrama de flujo de control que ilustra un método de codificación de información de vídeo de acuerdo con la presente invención; y

la Figura 5 es un diagrama de flujo de control que ilustra un método de decodificación de información de vídeo de acuerdo con la presente invención.

Modo para la invención

25 Se describen en detalle con referencia a los dibujos adjuntos algunas realizaciones ejemplares de la presente invención. Sin embargo, al describir las realizaciones de esta especificación, se omitirá una descripción detallada de funciones y constituciones muy conocidas si se considera que hará innecesariamente impreciso lo esencial de la presente invención.

30 En esta especificación, cuando se diga que un elemento está 'conectado' o 'acoplado' al otro elemento, puede significar que dicho elemento puede estar directamente conectado o acoplado al otro elemento o que un tercer elemento puede estar 'conectado' o 'acoplado' entre los dos elementos. Además, en esta especificación, cuando se diga que un elemento específico está 'incluido', puede significar que no estén excluidos elementos que no sean el elemento específico y que pueden estar incluidos elementos adicionales en las realizaciones ejemplares de la presente invención o el alcance técnico de la presente invención.

35 Pueden utilizarse términos, tales como el primer y el segundo, para describir diversos elementos, pero los elementos no están restringidos por los términos. Los términos se utilizan sólo para distinguir un elemento de otro elemento. Por ejemplo, un primer elemento puede denominarse segundo elemento sin apartarse del alcance de la presente invención. Asimismo, un segundo elemento puede denominarse primer elemento.

40 Además, las unidades de elemento descritas en las realizaciones ejemplares de la presente invención se muestran independientemente para indicar una diferencia y funciones características, y esto no significa que cada una de las unidades de elemento esté formada por una parte de *hardware* separado o una parte de *software*. Es decir que las unidades de elemento están dispuestas e incluidas para mayor comodidad de descripción, y al menos dos de las unidades de elemento pueden formar una unidad de elemento, o un elemento puede dividirse en una pluralidad de unidades de elemento y la pluralidad de unidades de elemento dividido pueden realizar funciones. En el alcance de la presente invención se incluye también una realización en la que los elementos estén integrados o realizaciones de las
45 que se hayan separado algunos elementos, a no ser que éstas se aparten de la esencia de la presente invención.

Además, en la presente invención, algunos elementos no son elementos esenciales para realizar funciones esenciales, sino que pueden ser elementos opcionales para mejorar sólo el rendimiento. La presente invención puede implementarse utilizando sólo elementos esenciales para implementar la esencia de la presente invención que no sean elementos utilizados para mejorar sólo el rendimiento, y en el alcance de la presente invención se incluye una estructura que incluya sólo elementos esenciales que no sean elementos opcionales utilizados para mejorar sólo el
50 rendimiento.

La Figura 1 es un diagrama de bloques que muestra un ejemplo de una estructura de un aparato de codificación de vídeo según una realización ejemplar. Un método o aparato de codificación/decodificación de vídeo escalable puede implementarse mediante una ampliación de un método o aparato de codificación/decodificación de vídeo común que no proporcione escalabilidad. El diagrama de bloques de la Figura 1 muestra una realización ejemplar de un aparato de codificación de vídeo que puede convertirse en una base para un aparato de codificación de vídeo escalable.

Remitiéndonos a la Figura 1, el aparato 100 de codificación de vídeo incluye un módulo 111 de predicción de movimiento, un módulo 112 de compensación de movimiento, un módulo 120 de intrapredicción, un conmutador 115, un substractor 125, un módulo 130 de transformación, un módulo 140 de cuantificación, un módulo 150 de codificación entrópica, un módulo 160 de descuantificación, un módulo 170 de transformación inversa, un sumador 175, un filtro 180 y una memoria intermedia 190 de imágenes de referencia.

El aparato 100 de codificación de vídeo puede realizar una codificación en una imagen de entrada en intramodo o intermodo y dar salida a un tren de bits como resultado de la codificación. En esta especificación, intrapredicción tiene el mismo significado que predicción intraimagen, e interpredicción tiene el mismo significado que predicción interimagen. En el caso del intramodo, el conmutador 115 puede conmutar a intramodo. En el caso del intermodo, el conmutador 115 puede conmutar a intermodo. El aparato 100 de codificación de vídeo puede generar un bloque predicho para el bloque de entrada de una imagen de entrada y luego codificar el remanente entre el bloque de entrada y el bloque predicho.

En el caso del intramodo, el módulo 120 de intrapredicción puede generar el bloque predicho realizando una predicción espacial utilizando valores de pixel de bloques cercanos, a un bloque actual, que ya estén codificados.

En el caso del intermodo, el módulo 111 de predicción de movimiento puede obtener un vector de movimiento buscando en una imagen de referencia, almacenada en la memoria intermedia 190 de imágenes de referencia, una zona que sea la más coincidente con el bloque de entrada en un proceso de estimación de movimiento. El módulo 112 de compensación de movimiento puede generar el bloque predicho realizando una compensación de movimiento utilizando el vector de movimiento y la imagen de referencia almacenada en la memoria intermedia 190 de imágenes de referencia.

El substractor 125 puede generar un bloque remanente sobre la base del remanente entre el bloque de entrada y el bloque predicho generado. El módulo 130 de transformación puede realizar una transformación en el bloque remanente y dar salida a un coeficiente de transformación según el bloque transformado. Además, el módulo 140 de cuantificación puede dar salida a un coeficiente cuantificado, cuantificando el coeficiente de transformación recibido utilizando un parámetro de cuantificación y/o una matriz de cuantificación.

El módulo 150 de codificación entrópica puede realizar una codificación entrópica en símbolos según una distribución de probabilidad sobre la base de valores calculados por el módulo 140 de cuantificación o valores de parámetro de codificación calculados en un proceso de codificación y dar salida a un tren de bits como resultado de la codificación entrópica. El método de codificación entrópica es un método para recibir símbolos que tienen diversos valores y representar los símbolos en forma de una cadena de números binarios decodificables eliminando al mismo tiempo la redundancia estadística de los símbolos.

Aquí, el símbolo se refiere a un elemento de sintaxis y un parámetro de codificación que se ha de codificar o decodificar, un valor de una señal remanente, etc. El parámetro de codificación es un parámetro necesario para la codificación y decodificación. El parámetro de codificación puede incluir no sólo información codificada mediante un codificador y luego comunicada a un decodificador junto con un elemento de sintaxis, sino también información que puede inducirse en un proceso de codificación o decodificación. El parámetro de codificación significa información necesaria para codificar o decodificar vídeo. El parámetro de codificación puede incluir, por ejemplo, un valor o estadísticas de modo intrapredicción/interpredicción, un vector de movimiento, un índice de imagen de referencia, un patrón de bloque de codificación, información sobre si está presente o no una señal remanente, un coeficiente de transformación, un coeficiente de transformación cuantificado, un parámetro de cuantificación, un tamaño de bloque, e información de división de bloque.

Además, la señal remanente puede significar una diferencia entre la señal original y una señal predicha. Además, la señal remanente puede significar una señal obtenida transformando una diferencia entre la señal original y una señal predicha o una señal obtenida transformando y cuantificando una diferencia entre la señal original y una señal predicha. La señal remanente puede denominarse bloque remanente en una unidad de bloque.

Si se utiliza codificación entrópica, el tamaño de un tren de bits para un símbolo que se haya de codificar puede reducirse, porque el símbolo se representa asignando un número pequeño de bits a un símbolo que tenga una alta frecuencia de aparición y un número grande de bits a un símbolo que tenga una baja frecuencia de aparición. Por consiguiente, mediante la codificación entrópica es posible mejorar el rendimiento de compresión de la codificación de vídeo.

Para la codificación entrópica pueden utilizarse métodos tales como Golomb exponencial, codificación de longitud variable adaptativa según el contexto (CAVLC, por sus siglas en inglés) y codificación aritmética binaria adaptativa

- según el contexto (CABAC, por sus siglas en inglés). Por ejemplo, en el módulo 150 de codificación entrópica puede estar almacenada una tabla para realizar una codificación entrópica, tal como una tabla de codificación/código de longitud variable (VLC, por sus siglas en inglés), y el módulo 150 de codificación entrópica puede realizar una codificación entrópica utilizando la tabla VLC almacenada. Además, el módulo 150 de codificación entrópica puede obtener un método de binarización para un símbolo objetivo y un modelo de probabilidad para un *bin*/símbolo objetivo y realizar una codificación entrópica utilizando el método de binarización o el modelo de probabilidad obtenido.
- El coeficiente cuantificado se descuantifica mediante el módulo 160 de descuantificación y luego se transforma inversamente mediante el módulo 170 de transformación inversa. El coeficiente descuantificado e inversamente transformado puede sumarse al bloque predicho a través del sumador 175, generando así un bloque reconstruido.
- El bloque reconstruido experimenta el filtro 180. El filtro 180 puede aplicar al bloque reconstruido o a la imagen reconstruida uno o más de los siguientes: un filtro de desbloqueo, una compensación adaptativa de muestra (SAO, por sus siglas en inglés) y un filtro de bucle adaptativo (ALF, por sus siglas en inglés). El bloque reconstruido que ha experimentado el filtro 180 puede almacenarse en la memoria intermedia 190 de imágenes de referencia.
- La Figura 2 es un diagrama de bloques que muestra un ejemplo de una estructura de un aparato de decodificación de vídeo según una realización ejemplar. Como se ha descrito anteriormente con referencia a la Figura 1, un método o aparato de codificación/decodificación de vídeo escalable puede implementarse mediante una ampliación de un método o aparato de codificación/decodificación común que no proporcione escalabilidad. El diagrama de bloques de la Figura 2 muestra una realización ejemplar de un aparato de decodificación de vídeo que puede convertirse en una base para un aparato de decodificación de vídeo escalable.
- Remitiéndonos a la Figura 2, el aparato 200 de decodificación de vídeo incluye un módulo 210 de decodificación entrópica, un módulo 220 de cuantificación inversa, un módulo 230 de transformación inversa, un módulo 240 de intrapredicción, un módulo 250 de compensación de movimiento, un filtro 260 y una memoria intermedia 270 de imágenes de referencia.
- El aparato 200 de decodificación de vídeo puede recibir un tren de bits que haya salido de un codificador, realizar una decodificación del tren de bits en un intramodo o intermodo y dar salida a una imagen reconstruida, es decir una imagen reconstruida. En el caso del intramodo, un conmutador puede conmutar a intramodo. En el caso del intermodo, el conmutador puede conmutar a intermodo. El aparato 200 de decodificación de vídeo puede obtener un bloque remanente reconstruido del tren de bits recibido, generar un bloque predicho y luego generar un bloque reconstruido, es decir un bloque reconstruido sumando el bloque remanente reconstruido al bloque predicho.
- El módulo 210 de decodificación entrópica puede generar símbolos que incluyen un símbolo que tiene forma de un coeficiente cuantificado, realizando una decodificación entrópica en el tren de bits recibido según una distribución de probabilidad. El método de decodificación entrópica es un método para recibir una cadena de números binarios y generar cada símbolo utilizando la cadena de números binarios. El método de decodificación entrópica es similar al método de codificación entrópica descrito anteriormente.
- El coeficiente cuantificado se descuantifica mediante el módulo 220 de cuantificación inversa y se transforma inversamente mediante el módulo 230 de transformación inversa. Como resultado de la descuantificación/transformación inversa del coeficiente cuantificado puede generarse un bloque remanente.
- En el caso del intramodo, el módulo 240 de intrapredicción puede generar un bloque predicho realizando una predicción espacial utilizando valores de píxel de bloques ya decodificados cercanos al bloque actual. En el caso del intermodo, el módulo 250 de compensación de movimiento puede generar un bloque predicho realizando una compensación de movimiento utilizando un vector de movimiento y una imagen de referencia almacenada en la memoria intermedia 270 de imágenes de referencia.
- El bloque remanente y el bloque predicho se suman mediante un sumador 255. El bloque sumado experimenta el filtro 260. El filtro 260 puede aplicar al bloque reconstruido o a la imagen reconstruida al menos uno de los siguientes: un filtro de desbloqueo, una SAO y un ALF. El filtro 260 da salida a una imagen reconstruida, es decir una imagen reconstruida. La imagen reconstruida puede almacenarse en la memoria intermedia 270 de imágenes de referencia y puede utilizarse para una predicción intertrama.
- De entre el módulo 210 de decodificación entrópica, el módulo 220 de cuantificación inversa, el módulo 230 de transformación inversa, el módulo 240 de intrapredicción, el módulo 250 de compensación de movimiento, el filtro 260 y la memoria intermedia 270 de imágenes de referencia incluidos en el aparato 200 de decodificación de vídeo, los elementos directamente relacionados con la decodificación de vídeo, por ejemplo el módulo 210 de decodificación entrópica, el módulo 220 de cuantificación inversa, el módulo 230 de transformación inversa, el módulo 240 de intrapredicción, el módulo 250 de compensación de movimiento y el filtro 260, pueden representarse como un módulo de decodificación con el fin de distinguirlos de otros elementos.
- Además, el aparato 200 de decodificación de vídeo puede incluir adicionalmente un módulo de análisis sintáctico (no mostrado) para analizar sintácticamente información relacionada con vídeo codificado incluida en un tren de bits. El

módulo de análisis sintáctico puede incluir el módulo 210 de decodificación entrópica, o el módulo de análisis sintáctico puede estar incluido en el módulo 210 de decodificación entrópica. El módulo de análisis sintáctico puede representarse como uno de los elementos del módulo de decodificación.

5 La Figura 3 es un diagrama conceptual que muestra esquemáticamente una realización ejemplar de una estructura de codificación de vídeo escalable (SVC) que utiliza múltiples capas a la que puede aplicarse la presente invención. En la Figura 3, las siglas GOP (*Group of Picture*) indican un grupo de imágenes.

Para enviar datos de vídeo es necesario un medio de transmisión, y un medio de transmisión tiene un rendimiento diferente dependiendo de diversos entornos de red. Para aplicaciones a los diversos medios de transmisión o entornos de red, puede emplearse un método de codificación de vídeo escalable (SVC).

10 El método SVC es un método de codificación para mejorar el rendimiento de codificación/decodificación eliminando la redundancia entre capas utilizando información de textura, información de movimiento, una señal remanente, etc. entre las capas. El método SVC puede proporcionar diversas escalabilidades desde los puntos de vista espacial, temporal y de relación señal-ruido (SNR) dependiendo de las condiciones circundantes, tales como una velocidad de transmisión de bits, una tasa de errores de transmisión y los recursos del sistema.

15 La SVC puede realizarse utilizando una estructura multicapa, de manera que pueda proporcionarse un tren de bits aplicable a diversas condiciones de red. Por ejemplo, una estructura de SVC puede incluir una capa base cuyos datos de vídeo puedan comprimirse y procesarse utilizando un método de codificación de vídeo común y puede incluir una capa de mejora cuyos datos de vídeo puedan comprimirse y procesarse utilizando tanto información de codificación de la capa base como un método de codificación de vídeo común.

20 Aquí, una capa significa un conjunto de imágenes y trenes de bits clasificados sobre la base de una resolución espacial (por ejemplo un tamaño de imagen), una resolución temporal (por ejemplo orden de codificación, orden de salida de las imágenes y una frecuencia de imagen), SNR y complejidad. Además, la capa base puede significar una capa de referencia, y la capa de mejora puede significar una capa de mejora. Además, múltiples capas pueden tener dependencia entre sí.

25 Remitiéndonos a la Figura 3, por ejemplo, una capa base puede estar definida por una definición estándar (SD, por sus siglas en inglés), una frecuencia de imagen de 15 Hz y una velocidad de transmisión de bits de 1 Mbps. Una primera capa de mejora puede estar definida por una alta definición (HD), una frecuencia de imagen de 30 Hz y una velocidad de transmisión de bits de 3,9 Mbps. Una segunda capa de mejora puede estar definida por una ultra alta definición (UHD) 4K, una frecuencia de imagen de 60 Hz y una velocidad de transmisión de bits de 27,2 Mbps. Un formato, una frecuencia de imagen, una velocidad de transmisión de bits, etc. son sólo realizaciones ejemplares y pueden estar determinados de un modo diferente si es necesario. Además, el número de capas utilizado no está limitado a la presente realización ejemplar y puede estar determinado de un modo diferente según las circunstancias.

30 Por ejemplo, si un ancho de banda de transmisión es de 4 Mbps, la frecuencia de imagen de la primera capa HD de mejora puede reducirse a menos de 15 Hz. El método SVC puede proporcionar escalabilidades temporales, espaciales y de SNR según el método descrito anteriormente en relación con la realización ejemplar de la Figura 3.

SVC tiene el mismo significado que codificación de vídeo escalable desde el punto de vista de la codificación y tiene el mismo significado que decodificación de vídeo escalable desde el punto de vista de la decodificación.

40 Como se ha descrito anteriormente, la escalabilidad se ha convertido actualmente en una importante función de un formato de vídeo debido a las redes de comunicación heterogéneas y a una diversidad de terminales. La SVC, es decir un estándar de ampliación de la codificación de vídeo avanzada (AVC, por sus siglas en inglés), se desarrolló para generar un tren de bits que tuviera una amplia gama de velocidades de transmisión de bits, manteniendo al mismo tiempo el máximo grado de eficacia de compresión. Para satisfacer las características y los cambios de diversos dispositivos y redes, un tren de bits SVC puede extraerse fácilmente de diversas formas. Es decir que el estándar SVC proporciona escalabilidad espacial, temporal y de SNR.

45 Entretanto, un tren de bits que incluye múltiples capas consta de unidades de capa de abstracción de red (NAL) que permiten el transporte adaptativo de vídeo a través de una red con conmutación de paquetes. Como en una multicapa, una relación entre una pluralidad de vistas en una codificación de vídeo multivista que incluya una pluralidad de vídeos multivista dentro de un tren de bits es similar a una relación entre capas espaciales en vídeo que soporte múltiples capas.

50 Para transformar un tren de bits eficazmente y eficientemente en todos los nodos en una ruta de distribución de contenido, la información de escalabilidad sobre el tren de bits es muy importante. En un estándar actual para la codificación de vídeo para una sola capa (es decir codificación de vídeo de alta eficiencia), en un encabezamiento de unidad de NAL (denominada en lo que sigue 'NALU') están presentes dos campos relacionados con la información de capa, es decir temporal_id y reserved_one_5bits. El campo 'temporal_id', que tiene una longitud de 3 bits, indica la capa temporal de un tren de bits de vídeo, y el campo 'reserved_one_5bits' corresponde a un área para indicar otra información de capa subsiguiente.

La capa temporal significa la capa de un tren de bits temporalmente escalable que incluye una NALU de capa de codificación de vídeo (VCL, por sus siglas en inglés), y la capa temporal tiene un valor `temporal_id` específico.

5 La presente invención se refiere a un método para describir eficazmente información de extracción e información de escalabilidad sobre una imagen dentro de un tren de bits que soporta múltiples capas y señalar la información de extracción y la información de escalabilidad, y a un aparato para implementar el método.

En la presente invención, un tren de bits está dividido en dos tipos: un tipo base que soporta sólo escalabilidad temporal y un tipo ampliado capaz de tener escalabilidad que soporta espacial/SNR/multivista.

10 El primer tipo de tren de bits se refiere a un tren de bits que soporta un vídeo de una sola capa, y el segundo tipo del mismo se refiere a una capa de mejora en una codificación de vídeo en capas basada en HEVC. A continuación, se propone un método mejorado para representar información de escalabilidad sobre los dos tipos de tren de bits. De acuerdo con la presente invención, en el tipo ampliado, puede utilizarse 'reserved_one_5bits' de 5 bits como `layer_id` indicativo del identificador de una capa escalable.

Eliminación de `nal_ref_flag` del encabezamiento NALU

15 `nal_ref_flag` se utiliza para indicar una imagen no de referencia. Esta información indica una prioridad aproximada entre una imagen no de referencia y una imagen de referencia, pero el uso de `nal_ref_flag` está algo limitado.

Una imagen de referencia significa una imagen que incluye muestras que pueden utilizarse para una predicción interimagen cuando se decodifican imágenes subsiguientes en el orden de decodificación.

Una imagen no de referencia significa una imagen que incluye muestras no utilizadas para la predicción interimagen cuando se decodifican imágenes subsiguientes en el orden de decodificación.

20 `nal_ref_flag` es un indicativo de etiqueta de información que indica si una NALU correspondiente es una imagen no de referencia o una imagen de referencia en la totalidad del tren de bits en el momento de la codificación.

25 Cuando un valor de `nal_ref_flag` es 1, una NALU significa que incluye un conjunto de parámetros de secuencia (SPS, por sus siglas en inglés), un conjunto de parámetros de imagen (PPS, por sus siglas en inglés), un conjunto de parámetros de adaptación (APS, por sus siglas en inglés), o la porción (*slice*) de una imagen de referencia. Cuando un valor de `nal_ref_flag` es 0, una NALU significa que incluye una porción que incluye parte o la totalidad de la imagen no de referencia.

30 Aquí, una NALU en la que un valor de `nal_ref_flag` sea 1 puede incluir la porción de una imagen de referencia. `nal_ref_flag` tiene un valor de 1 para las NALU de un conjunto de parámetros de vídeo (VPS, por sus siglas en inglés), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de imagen (PPS). Si un valor de `nal_ref_flag` es 0 en una de las NALU de VCL de una imagen específica, `nal_ref_flag` tiene un valor de 0 para todas las NALU de VCL de la imagen.

Entretanto, si se extraen todas las imágenes de no referencia, en particular imágenes de no referencia correspondientes principalmente a la capa temporal superior, el `nal_ref_flag` de todas las imágenes restantes después de la extracción adopta el valor 1.

35 Sin embargo, algunas imágenes de un tren de bits transformado de manera adaptativa, es decir imágenes correspondientes a la capa temporal superior en los trenes de bits restantes, pasan a ser imágenes de no referencia aunque un valor de `nal_ref_flag` sea 1.

40 En otras palabras, otro elemento de sintaxis (por ejemplo `temporal_id`) de un encabezamiento de NALU puede ser más eficaz a la hora de soportar una transformación adaptativa (o extracción). Es decir que un tren de bits que incluya una capa temporal deseada puede extraerse utilizando el número total de capas temporales que incluyan un tren de bits y un valor de `temporal_id` de un encabezamiento de NALU.

45 Además, `nal_ref_flag` también puede utilizarse para indicar si una imagen correspondiente se utilizará o no posteriormente como imagen de referencia a la hora de decodificar (reconstruir) una imagen formada por una NALU que incluya `nal_ref_flag` y luego almacenar la imagen decodificada en una memoria, tal como una memoria intermedia de imágenes decodificadas (DPB, por sus siglas en inglés). Si el valor de `nal_ref_flag` es 1, indica que una imagen correspondiente se utilizará posteriormente como imagen de referencia. Si el valor de `nal_ref_flag` es 0, indica que una imagen correspondiente no se utilizará posteriormente como imagen de referencia.

50 Una imagen decodificada puede indicarse como una imagen de referencia a la hora de almacenar la imagen decodificada en la DPB sin determinar si una NALU correspondiente es o no una imagen de no referencia o una imagen de referencia sobre la base de `nal_ref_flag`. En este caso, aunque la imagen decodificada sea una imagen de no referencia, pero esté indicada como una imagen de referencia, no hay problema alguno porque la imagen correspondiente no se incluirá en la lista de imágenes de referencia entregada en el encabezamiento de porción de

una imagen siguiente a la hora de decodificar la imagen siguiente a la imagen correspondiente en el orden de decodificación.

5 Es decir que el que una imagen previamente decodificada sea o no una imagen de referencia o una imagen de no referencia se indica sobre la base de la lista de imágenes de referencia incluida en el encabezamiento de porción de una imagen siguiente, a la hora de decodificar la imagen siguiente. Por consiguiente, no hay problema alguno en determinar si una imagen decodificada es o no una imagen de referencia o una imagen de no referencia, aunque la imagen decodificada se indique como la imagen de referencia sin tener en cuenta `nal_ref_flag`.

10 La presente invención propone que se borre `nal_ref_flag` de un encabezamiento de NALU o que se cambie la semántica de `nal_ref_flag`. Una realización relacionada con el borrado de `nal_ref_flag` es como se indica a continuación.

Realización 1

La etiqueta '`nal_ref_flag`' se cambia a `slice_ref_flag`, y la posición de la etiqueta se mueve de un encabezamiento de NALU a un encabezamiento de porción. La sintaxis del encabezamiento de porción puede modificarse como en la Tabla 1.

15 Tabla 1

<code>slice_header() {</code>	Descriptor
<code>slice_ref_flag</code>	<code>u(1)</code>
<code>first_slice_in_pic_flag</code>	<code>u(1)</code>
<code>if (first_slice_in_pic_flag == 0)</code>	
<code>slice_address</code>	<code>u(v)</code>
<code>...</code>	<code>ue(v)</code>
<code>}</code>	

En la Tabla 1, cuando un valor de `slice_ref_flag` es 1, indica que una porción es parte de una imagen de referencia. Cuando un valor de `slice_ref_flag` es 0, indica que la porción es parte de una imagen de no referencia.

Realización 2

20 La etiqueta '`nal_ref_flag`' se cambia a `au_ref_flag`, y la posición de la etiqueta se mueve de un encabezamiento de NALU a un delimitador de unidad de acceso. La sintaxis del delimitador de unidad de acceso puede ser la misma que en la Tabla 2.

Tabla 2

<code>access_unit_delimiter_rbsp() {</code>	Descriptor
<code>pic_type</code>	<code>u(3)</code>
<code>au_ref_flag</code>	<code>u(1)</code>
<code>rbp_trailing_bits()</code>	
<code>}</code>	

25 En la Tabla 2, cuando un valor de `au_ref_flag` es 1, indica que una unidad de acceso incluye una imagen de referencia. Cuando un valor de `au_ref_flag` es 0, indica que una unidad de acceso incluye una imagen de no referencia.

Realización 3

No se mueve `nal_ref_flag` a otra sintaxis, pero se borra `nal_ref_flag` de un encabezamiento de NALU.

- 5 Si se borra `nal_ref_flag`, es decir la información de etiqueta de 1 bit que indica si una imagen es una imagen no de referencia o una imagen de referencia en la totalidad del tren de bits a la hora de decodificar la imagen, puede realizarse una determinación de si una imagen es o no una imagen de referencia a través de `nal_ref_flag` a través de otro proceso. Después de decodificar una imagen recibida, la imagen decodificada se indica incondicionalmente como una imagen de referencia en una memoria intermedia de imágenes decodificadas (DPB). Es decir que puede no determinarse si una imagen decodificada es o no una imagen de referencia, pero la imagen decodificada puede indicarse como una imagen de referencia.
- 10 Después, se analiza sintácticamente el encabezamiento de porción de una imagen siguiente a la imagen decodificada, y el que la imagen decodificada sea una imagen de referencia o una imagen no de referencia puede indicarse sobre la base de información de imagen de referencia incluida en el encabezamiento de porción.

Realización 4

- 15 Puede borrarse `nal_ref_flag` de un encabezamiento de NALU y puede utilizarse `temporal_id` para indicar que la NALU es una imagen no de referencia. Para indicar una imagen no de referencia, `temporal_id` puede ser '7', un número máximo de capas temporales-1 (es decir `max_temporal_layers_minus1`) incluido en un tren de bits, o un valor prefijado distinto de '0'.

Realización 5

- 20 Puede borrarse `nal_ref_flag` de un encabezamiento de NALU y puede utilizarse `reserved_one_5bits` como elemento `priority_id` para indicar que la NALU es una imagen no de referencia. `priority_id` es un identificador que indica una prioridad de la NALU correspondiente y se utiliza para proporcionar una función de extracción de tren de bits sobre la base de una prioridad, sin tener en cuenta una espacial, un temporal o una SNR diferente.

Es decir que si `temporal_id = Ta` es el identificador de la capa temporal superior, se utiliza `temporal_id = Ta` y la NALU, es decir `priority_id = 31` (u otro valor específico), para indicar que la NALU es la NALU de una imagen no de referencia.

- 25 1 bit utilizado para señalar `nal_ref_flag` puede utilizarse como una de las siguientes cosas.
- (1) Dicho 1 bit puede utilizarse para indicar `nal_unit_type`. `nal_unit_type` puede convertirse en una señal de 7 bits y puede duplicarse el número de tipos de NALU.
- (2) Dicho 1 bit puede utilizarse para indicar `temporal_id`. `temporal_id` puede convertirse en una señal de 4 bits y puede duplicarse un número máximo de capas temporales.
- 30 (3) Dicho 1 bit puede utilizarse para indicar `layer_id`. `layer_id` significa el identificador de la capa escalable de un tren de bits en capas y puede señalizarse mediante un elemento de sintaxis `reserved_one_5bits`. 1 bit utilizado para señalar `nal_ref_flag` puede sumarse a 5 bits de `reserved_one_5bits` utilizados para identificar una capa escalable y, por lo tanto, `layer_id` puede convertirse en una señal de 6 bits. Si se utilizan 6 bits, pueden identificarse 64 capas escalables.
- 35 (4) Dicho 1 bit puede utilizarse como una etiqueta que informe de si `reserved_one_5bits` indica o no prioridad.
- (5) Dicho 1 bit puede utilizarse como `reserved_bit`.

Si no se borra `nal_ref_flag` de un encabezamiento de NALU, la semántica de `nal_ref_flag` puede modificarse de la siguiente manera.

- 40 Cuando el valor de `nal_ref_flag` es 0, indica que una NALU incluye sólo la porción de una imagen no de referencia. Cuando el valor de `nal_ref_flag` es 1, indica que una NALU puede incluir la porción de una imagen de referencia o de una imagen no de referencia.

Señalización de activación de conjunto de parámetros de vídeo (VPS)

Un VPS incluye la información más básica para decodificar vídeo y puede incluir contenidos presentes en el SPS existente.

- 45 El VPS puede incluir información sobre una subcapa que indique una capa temporal que soporta escalabilidad temporal e información sobre múltiples capas que soporten una escalabilidad espacial, de calidad y de vista. Es decir que el VPS puede incluir información multicapa, es decir sintaxis para una ampliación de HEVC.

A. Conjunto de parámetros de vídeo

Las sintaxis para un VPS son iguales que en la Tabla 3.

Tabla 3

	Descriptor
video_parameter_set_rbsp() {	
video_parameter_set_id	ue(v)
max_temporal_layers_minus1	u(3)
priority_id_flag	u(1)
extension_info_flag	u(1)
if(extension_info_flag == 0) {	
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_cropping_flag	u(1)
if(pic_cropping_flag) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
chroma-format-idc	u(2)
temporal_id_nesting_flag	u(1)
for (i = 0; i <= num_temporal_layers_minus1; i++) {	
profile_idc[i]	u(8)
reserved_zero_8_bits[i] /* equal to 0 */	u(8)
level_idc[i]	u(8)
}	
tiles_or_entropy_coding_sync_idc	u(2)

if(tiles_or_entropy_coding_sync_idc == 1){	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag){	
for (i= 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
}	
if(extension_info_flag == 1){	
extension_info()	
}	
rbsp_trailing_bits()	
}	

En la Tabla 3, la mayoría de las sintaxis tienen la misma semántica que las sintaxis de SPS aplicadas a un tren de bits que incluye una sola capa, y las partes adicionales son como se indica a continuación.

5 - video_parameter_set_id significa el identificador de un VPS y puede hacerse referencia a video_parameter_set_id en un conjunto de parámetros de secuencia (SPS), información de mejora suplementaria (SEI, por sus siglas en inglés) o delimitadores de unidad de acceso.

- Cuando el valor de priority_id_flag es 1, indica que reserved_one_5bits puede utilizarse idénticamente con priority_id de un estándar SVC. Cuando el valor de priority_id_flag es 0, significa que reserved_one_5bits puede utilizarse como layer_id.

10 - Cuando el valor de extension_info_flag es 0, indica que un tren de bits cumple un estándar de una sola capa de HEVC. Cuando el valor de extension_info_flag es 1, indica una capa de mejora para soportar escalabilidad (es decir cuando se soporta una ampliación de HEVC) y se proporciona información relacionada con una capa.

B. Modificación de conjunto de parámetros de secuencia (SPS)

15 Como en la Tabla 4, parte de una sintaxis existente puede incorporarse a un VPS y puede borrarse de un SPS. Entretanto, puede añadirse al SPS un elemento de sintaxis vps_id. Una sintaxis de SPS a la que se le haya añadido vps_id es igual que en la Tabla 4. En la Tabla 4, una sintaxis borrada se indica mediante una línea que pasa por el centro de la sintaxis.

vps_id indica un identificador para identificar un VPS al que puede hacerse referencia en el SPS y puede tener un margen de 0 a X.

Tabla 4

	Descriptor
seq_parameter_set_rbsp() {	
seq_parameter_set_id	ue(v)
vps_id	ue(v)
...	
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

5 C. Señalización de activación para VPS

Un encabezamiento de porción incluye información de índice sobre un conjunto de parámetros de imagen (PPS) al que se refiere una porción correspondiente, y un PPS incluye información de índice sobre un conjunto de parámetros de secuencia (SPS) al que se refiere una imagen correspondiente. El SPS incluye información sobre un conjunto de parámetros de vídeo (VPS) al que se refiere una secuencia correspondiente. Como se ha descrito anteriormente, cuando se analiza sintácticamente información sobre un conjunto de parámetros y, luego, se hace referencia a información sobre el conjunto de parámetros analizado sintácticamente, se denomina activación.

Para utilizar información sobre un conjunto de parámetros específico, es decir para activar el conjunto de parámetros, es necesario analizar sintácticamente de forma gradual el conjunto de parámetros a partir de un encabezamiento de porción. Esto significa que es necesario analizar todos los encabezamientos de porción y un PPS relacionado para saber qué SPS se activa.

Cuando se extrae parte de una subcapa (es decir capa temporal) de un tren de bits que incluye una sola capa, un extractor necesita analizar (o analizar sintácticamente) un encabezamiento de NALU y una pluralidad de conjuntos de parámetros.

Si en un VPS o un SPS está incluida información para la extracción de una NALU, el extractor necesita analizar sintácticamente de forma secuencial un conjunto de parámetros mayor a partir de un encabezamiento de porción. Esto significa que el extractor necesita entender todos los elementos de sintaxis de los conjuntos de parámetros y el encabezamiento de porción.

Por otra parte, sin un proceso de análisis sintáctico complicado incluso en un proceso de decodificación de vídeo, puede buscarse vps_id o sps_id y pueden activarse sólo los conjuntos de parámetros necesarios. En este caso, si un VPS o un SPS incluye información de índice de parámetro que se haya de activar, puede reducirse un procedimiento de análisis sintáctico para un encabezamiento de porción complicado y un PPS relacionado.

Entretanto, sólo algunos de los elementos de las sintaxis pueden incluir datos necesarios para extraer un tren de bits. Sin embargo, analizar todos los elementos de sintaxis puede convertirse en una gran carga en un extractor. Para solucionar este problema, se propone el método siguiente.

En la presente invención, la activación de un conjunto de parámetros significa que se realiza una señalización de manera que un extractor pueda saber qué conjunto de parámetros se activa sin analizar un encabezamiento de porción y un conjunto de parámetros de imagen (PPS) relacionado.

5 De acuerdo con la presente invención, puede señalizarse adicionalmente qué VPS, SPS o PPS está activo, de manera que se reduzca una carga en un extractor que necesite analizar todos los encabezamientos de porción y un PPS relacionado.

Un VPS puede actualizarse. Puede utilizarse uno de los siguientes tres métodos, de manera que un extractor pueda estar al corriente de un VPS activo y un SPS o PPS relacionado sin analizar un encabezamiento de porción.

10 (1) *vps_id*, *sps_id* y *pps_id* pueden incluirse en un delimitador de unidad de acceso. *vps_id*, *sps_id* y *pps_id* indican los identificadores de los VPS, SPS y PPS respectivos utilizados para NALU de una AU relacionada.

Para indicar si los identificadores están o no presentes en el delimitador de unidad de acceso se utilizan *vps_id_present_flag*, *sps_id_present_flag* y *pps_id_present_flag*. La sintaxis de un delimitador de unidad de acceso propuesto es la misma que en la Tabla 5.

Tabla 5

<code>access_unit_delimiter_rbsp() {</code>	Descriptor
pic_type	u(3)
vps_id_present_flag	u(1)
sps_id_present_flag	u(1)
pps_id_present_flag	u(1)
if (<i>vps_id_present_flag</i>)	
vps_id	ue(v)
if (<i>sps_id_present_flag</i>)	
sps_id	ue(v)
if (<i>pps_id_present_flag</i>)	
pps_id	ue(v)
<code> </code> <i>rbp_trailing_bits</i> <code>()</code>	
<code>}</code>	

15 (1-1) En otro método, se excluyen *sps_id* y *pps_id* y puede incluirse sólo *vps_id* en un delimitador de unidad de acceso como en la Tabla 6.

Tabla 6

<code>access_unit_delimiter_rbsp() {</code>	Descriptor
pic_type	u(3)
vps_id	ue(v)

rbsp_trailing_bits()	
}	

5 (2) Otro método para la señalización de activación de un VPS es utilizar un nuevo mensaje de SEI 'parameter_set_reference'. El mensaje de SEI incluye una sintaxis para informar de si están o no presentes vps_id, sps_id y pps_id indicativos de los identificadores de un VPS, un SPS y un PPS utilizados para NALU dentro de una AU relacionada.

Para indicar si los identificadores están o no presentes, puede utilizarse una sintaxis de vps_id_present_flag, sps_id_present_flag y pps_id_present_flag, y una sintaxis de SEI es igual que en la Tabla 7.

Tabla 7

parameter_set_reference(payloadSize) {	Descriptor
vps_id_present_flag	u(1)
sps_id_present_flag	u(1)
pps_id_present_flag	u(1)
if(vps_id_present_flag)	
vps_id	ue(v)
if(sps_id_present_flag)	
sps_id	ue(v)
if(pps_id_present_flag)	
pps_id	ue(v)
}	

10 (2-1) Además, puede informarse de la activación de un VPS y un SPS excluyendo pps_id e incluyendo sps_id y vps_id en un mensaje de SEI como en la Tabla 8. sps_id y vps_id en un mensaje de SEI pueden incluir sps_id y vps_id a los que se refiera la NALU de capa de codificación de vídeo de una unidad de acceso asociada con el mensaje de SEI. Por consiguiente, sps_id y vps_id pueden indicar información sobre un conjunto de parámetros que tenga una posibilidad de activación.

15 Tabla 8

parameter_set_reference(payloadSize) {	Descriptor
vps_id	u(4)
sps_id_present_flag	u(1)
if(sps_id_present_flag)	

sps_id	ue(v)
psr_extension_flag	u(1)
if(psr_extension_flag){	
psr_extension_length	ue(v)
for(i=0; i<psr_extension_length; i++)	
psr_extension_data_byte	u(8)
}	
}	

En la Tabla 8, vps_id indica video_parameter_set_id de un VPS ahora activado. Un valor de vps_id puede tener un valor de 0~15.

5 Si sps_id_present_flag tiene un valor de 1, indica que sequence_parameter_set_id de un SPS ahora activado está incluido en un mensaje de SEI correspondiente. Si sps_id_present_flag tiene un valor de 0, indica que sequence_parameter_set_id de un SPS ahora activado no está incluido en un mensaje de SEI correspondiente.

sps_id indica sequence_parameter_set_id de un SPS ahora activado. sps_id puede tener un valor de 0~31, de forma más limitada un valor de 0~15.

10 Cuando el valor de psr_extension_flag es 0, indica que un elemento de sintaxis de ampliación de mensaje de SEI de referencia de conjunto de parámetros no está incluido en un mensaje de SEI de referencia de conjunto de parámetros. Cuando el valor de psr_extension_flag es 1, indica que un elemento de sintaxis de ampliación de mensaje de SEI de referencia de conjunto de parámetros que incluye un mensaje de SEI de referencia de conjunto de parámetros está ampliado y es utilizado.

15 psr_extension_length indica la longitud de psr_extension_data. psr_extension_length puede tener un valor que oscila entre 0 y 256, y psr_extension_data_byte puede tener cualquier valor.

(2-2) Además, en un mensaje de SEI pueden incluirse uno o más sps_id y vps_id distintos de pps_id, que luego pueden señalizarse como en la Tabla 9.

Tabla 9

parameter_set_reference(payloadSize) {	Descriptor
vps_id	u(4)
num_referenced_sps	u(5)
for (i=0; i< num_referenced_sps; i++)	
sps_id(i)	ue(v)

}	

En la Tabla 9, vps_id indica video_parameter_set_id de un VPS activo. vps_id puede tener un valor de 0~15.

num_reference_sps indica el número de SPS que se refieren a vps_id activos.

5 sps_id(i) indica sequence_parameter_set_id de un SPS activo, y sps_id puede tener un valor de 0~31, de forma más limitada un valor de 0~15.

(2-3) Además, en un mensaje de SEI pueden incluirse sólo vps_id distintos de sps_id y pps_id, que luego pueden señalizarse como en la Tabla 10.

Tabla 10

parameter_set_reference(payloadSize){	Descriptor
vps_id	ue(v)
}	

10 (3) Otro método para la señalización de activación de un VPS es incluir información, que informe de vps_id, sps_id y pps_id, en un mensaje de SEI de periodo de almacenamiento en la memoria intermedia. La Tabla 11 muestra una sintaxis que incluye vps_id_present_flag, sps_id_present_flag y pps_id_present_flag que indican si están o no presentes vps_id, sps_id y pps_id.

Tabla 11

buffering_period(payloadSize) {	Descriptor
vps_id_present_flag	u(1)
sps_id_present_flag	u(1)
pps_id_present_flag	u(1)
if(vps_id_present_flag)	
vps_id	ue(v)
if(sps_id_present_flag)	
sps_id	ue(v)
if(pps_id_present_flag)	
pps_id	ue(v)

.....	
}	

(3-1) Además, como se indica en la Tabla 12, la activación de un VPS puede señalizarse incluyendo sólo vps_id distintos de sps_id y pps_id en el mensaje de SEI de periodo de almacenamiento en la memoria intermedia.

Tabla 12

	Descriptor
buffering_period(payloadSize) {	
vps_id_present_flag	u(1)
if(vps_id_present_flag)	
vps_id	ue(v)
.....	
}	

5

(4) Otro método para la señalización de activación de un conjunto de parámetros es incluir información, que informe de vps_id, sps_id y pps_id, en un mensaje de SEI de punto de recuperación. La Tabla 13 muestra una sintaxis que incluye vps_id_present_flag, sps_id_present_flag y pps_id_present_flag que indican si están o no presentes vps_id, sps_id y pps_id.

10 Tabla 13

	Descriptor
recovery_point(payloadSize) {	
vps_id_present_flag	u(1)
sps_id_present_flag	u(1)
pps_id_present_flag	u(1)
if(vps_id_present_flag)	
vps_id	ue(v)
if(sps_id_present_flag)	
sps_id	ue(v)
if(pps_id_present_flag)	
pps_id	ue(v)
recovery_poc_cnt	ue(v)
exact_match_flag	u(1)
broken_link_flag	u(1)

}	
recovery_point(payloadSize){	Descriptor
vps_id_present_flag	u(1)
if(vps_id_present_flag)	
vps_id	ue(v)
recovery_poc_cnt	ue(v)
exact_match_flag	u(1)
broken_link_flag	u(1)
}	

(4-1) Además, como se indica en la Tabla 14, puede haber un método para informar de vps_id, sps_id y pps_id incluyendo sólo vps_id distintos de sps_id y pps_id en el mensaje de SEI de punto de recuperación.

Tabla 14

recovery_point (payloadSize) {	Descriptor
vps_id_present_flag	u(1)
if (vps_id_present_flag)	
vps_id	ue(v)
recovery_poc_cnt	ue(v)
exact_match_flag	u(1)
broken_link_flag	u(1)
}	

5

Los mensajes para entregar vps_id o sps_id pueden incluirse en una unidad de acceso de punto de acceso aleatorio interno (IRAP, por sus siglas en inglés).

Si se incluye en una unidad de acceso y se utiliza cualquiera de los métodos de señalización de información anteriormente descritos, un extractor puede identificar valores vps_id, sps_id y pps_id a través del método de señalización anteriormente descrito para extraer un tren de bits y puede gestionar uno o más vps/sps/pps.

Además, un aparato de decodificación o un módulo de decodificación para realizar una decodificación puede identificar valores vps_id, sps_id y pps_id a través del método de señalización anteriormente descrito y, con el método de señalización, activando los conjuntos de parámetros, puede decodificar las AU asociadas.

Representación de un tren de bits en un tipo ampliado

15 A continuación se proponen extension_info() de VPS y un nuevo mensaje de SEI para describir y señalar información sobre una capa escalable si está incluido un tren de bits que soporte una ampliación de capa. Para representar un tren de bits en el tipo ampliado, puede señalizarse la siguiente información.

layer_id señala si entrega o no un valor de prioridad de una capa.

Aquí, una capa espacial (identificada mediante un valor dependency_id), una capa de SNR (identificada mediante un valor quality_id), vistas (identificadas mediante un valor view_id), etc. pueden señalizarse en respuesta a cada valor layer_id, y una capa temporal puede identificarse mediante temporal_id de un encabezamiento de NALU.

- 5 Además, la zona de vídeo relacionada con layer_id puede señalizarse mediante region_id.

Además, pueden señalizarse una información de dependencia, una información de velocidad de transmisión de bits para cada capa escalable y una información de calidad para cada capa escalable.

La sintaxis de extension_info es igual que en la Tabla 15.

Tabla 15

extension_info(){	
num_frame_sizes_minus1	ue(v)
num_rep_formats_minus1	ue(v)
for(i = 0; i <= num_frame_sizes_minus1; i++){	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
pic_cropping_flag[i]	u(1)
if(pic_cropping_flag[i]) {	
pic_crop_left_offset[i]	ue(v)
pic_crop_right_offsetv[i]	ue(v)
pic_crop_top_offset[i]	ue(v)
pic_crop_bottom_offset[i]	ue(v)
}	
}	
for(i = 0; i <= num_rep_formats_minus1; i++) {	
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
num_layers_minus1	ue(v)
dependency_id_flag	u(1)

quality_id_flag	u(1)
view_id_flag	u(1)
region_id_flag	u(1)
layer_dependency_info_flag	u(1)
for(i = 1; i <= num_layers_minus1; i++){	
frame_size_idx[i]	ue(v)
rep_format_idx[i]	ue(v)
if(dependency_id_flag) {	
one_dependency_id_flag[i]	u(1)
if(one_dependency_id_flag[i] == 1)	
dependency_id[i]	ue(v)
else if(one_dependency_id_flag[i] == 0){	
dependency_id_min[i]	ue(v)
dependency_id_max[i]	ue(v)
}	
}	
if(quality_id_flag){	
one_quality_id_flag[i]	u(1)
if(one_quality_id_flag[i] == 1)	
quality_id[i]	ue(v)
else if(one_quality_id_flag[i] == 0){	
quality_id_min[i]	ue(v)
quality_id_max[i]	ue(v)
}	
}	
if (view_id_flag) {	
one_view_id_flag[i]	u(1)

depth_flag[i]	u(1)
if (one_view_id_flag[i] == 1)	
view_id[i]	ue(v)
else if (one_view_id_flag[i] == 0) {	
view_id_min[i]	ue(v)
view_id_max[i]	ue(v)
}	
}	
if (region_id_flag) {	
num_regions_minus1[i]	u(8)
for (j = 1; j <= num_regions_minus1[i]; j++)	
region_id[i][j]	u(8)
}	
if (layer-dependency_info_flag){	
num_directly_dependent_layers[i]	u(8)
for (j = 0; j < num_directly_dependent_layers[i]; j++)	
directly_dependent_layer_id_delta_minus1[i][j]	u(8)
}	
}	
}	

La semántica de la sintaxis de la Tabla 15 es como se indica a continuación.

5 - num_frame_sizes_minus1 plus 1 indica una cantidad máxima de información de tamaño (por ejemplo pic_width_in_luma_samples[i], pic_height_in_luma_samples[i], pic_cropping_flag[i], pic_cropping_flag[i], pic_crop_left_offset[i], pic_crop_right_offsetv[i], pic_crop_top_offset[i] y pic_crop_bottom_offset[i]) de otros tipos de imágenes incluidas en una secuencia de vídeo codificada. El valor de num_frame_sizes_minus1 puede ser 0~X. Otros tipos de imágenes pueden incluir imágenes que tengan resoluciones diferentes.

10 - num_rep_formats_minus1 plus 1 indica un número máximo de tipos diferentes de profundidades de bits y formatos de crominancia (por ejemplo bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y valores[i] de chroma_format_idc) incluidos en una secuencia de vídeo codificada. El valor de num_rep_formats_minus1 puede estar en un intervalo de 0~X.

- pic_width_in_luma_samples[i], pic_height_in_luma_samples[i], pic_cropping_flag[i], pic_cropping_flag[i], pic_crop_left_offset[i], pic_crop_right_offsetv[i], pic_crop_top_offset[i] y pic_crop_bottom_offset[i] indican valores de orden i de pic_width_in_luma_samples, pic_height_in_luma_samples, pic_cropping_flag, pic_cropping_flag,

pic_crop_left_offset, pic_crop_right_offset, pic_crop_top_offset, pic_crop_bottom_offset de una secuencia de vídeo codificada.

- bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y chroma_format_idc[i] indican valores de orden i de bit_depth_luma_minus8, bit_depth_chroma_minus8 y chroma_format_idc de una secuencia de vídeo codificada.

- 5 - num_layers_minus1 indica el número de capas escalables disponibles en un tren de bits.
- Cuando el valor de dependency_id_flag es 1, indica que están presentes uno o más valores dependency_id relacionados con un valor layer_id.
- Cuando el valor de quality_id_flag es 1, indica que están presentes uno o más valores quality_id relacionados con un valor layer_id.
- 10 - Cuando el valor de view_id_flag es 1, indica que están presentes uno o más valores view_id relacionados con un valor layer_id.
- Cuando el valor de region_id_flag es 1, indica que están presentes uno o más valores region_id relacionados con un valor layer_id.
- 15 - Cuando el valor de layer_dependency_info_flag es 1, indica que se proporciona información de dependencia de una capa escalable.
- frame_size_idx[i] indica un índice para un conjunto de tamaños de trama aplicado a una capa que tiene un valor layer_id de i. frame_size_idx[i] tiene un valor que oscila entre 0 y X.
- rep_format_idx[i] indica un índice para un conjunto de una profundidad de bits y un formato de crominancia aplicado a una capa que tiene un valor layer_id de i. rep_format_idx[i] tiene un valor que oscila entre 0 y X.
- 20 - Cuando el valor de one_dependency_id_flag[i] es 1, indica que hay sólo un dependency_id asociado con una capa cuyo layer_id es i. Cuando el valor de one_dependency_id_flag[i] es 0, indica que hay dos o más dependency_id asociados con una capa cuyo layer_id es i.
- dependency_id[i] indica un valor de dependency_id asociado con una capa cuyo layer_id es i.
- dependency_id_min[i] y dependency_id_max[i] indican un valor dependency_id mínimo y un valor dependency_id máximo asociados con una capa cuyo layer_id es i, respectivamente.
- 25 - Cuando el valor de one_quality_id_flag[i] es 1, indica que hay sólo un quality_id asociado con una capa cuyo layer_id es i. Cuando el valor de one_quality_id_flag[i] es 0, indica que hay dos o más valores quality_id asociados con una capa cuyo layer_id es i.
- quality_id[i] indica un valor de quality id asociado con una capa cuyo layer_id es i.
- 30 - quality_id_min[i] y quality_id_max[i] indican un valor quality_id mínimo y un valor quality_id máximo asociados con una capa cuyo layer_id es i, respectivamente.
- Cuando el valor de one_view_id_flag[i] es 1, indica que hay un view_id asociado con una capa cuyo layer_id es i. Cuando el valor de one_view_id_flag[i] es 0, indica que hay dos o más valores view_id asociados con una capa cuyo layer_id es i.
- 35 - view_id[i] indica un valor de view_id asociado con una capa cuyo layer_id es i.
- Cuando el valor de depth_flag[i] es 1, indica que una capa escalable cuyo layer_id es i incluye información de profundidad de un tren de bits de vídeo 3-D.
- view_id_min[i] y view_id_max[i] indican un valor view_id mínimo y un valor view_id máximo asociados con una capa cuyo layer_id es i.
- 40 - num_regions_minus1 plus1 indica el número de zonas asociadas con una capa cuyo layer_id es i.
- region_id[j] indica el identificador de una zona j asociada con una capa cuyo layer_id es i.
- num_directly_dependent_layers[i] indica el número de capas escalables (es decir el número de capas necesarias para generar una señal predicha cuando se realiza una decodificación) con las que una capa escalable i actual está directamente asociada.
- 45 - directly_dependent_layer_id_delta_minus1[i][j] plus 1 indica una diferencia entre layer_id[i], es decir una capa escalable actual, y el identificador de capa de una capa escalable de orden j con la que la capa escalable actual está

directamente asociada. El identificador de capa de la capa escalable de orden j con la que la capa escalable actual está directamente asociada es igual que $(\text{layer_id}[i] - \text{directly_dependent_layer_id_delta_minus1}[i][j] - 1)$.

Una sintaxis de `extension_info()` según otra realización es igual que en la Tabla 16.

Tabla 16

<code>extension_info(){</code>	
<code> num_frame_sizes_minus1</code>	<code>ue(v)</code>
<code> num_rep_formats_minus1</code>	<code>ue(v)</code>
<code> for (i = 0; i <= num_rep_formats_minus1; i++) {</code>	
<code> pic_width_in_luma_samples[i]</code>	<code>ue(v)</code>
<code> pic_height_in_luma_samples[i]</code>	<code>ue(v)</code>
<code> bit_depth_luma_minus8[i]</code>	<code>ue(v)</code>
<code> bit_depth_chroma_minus8[i]</code>	<code>ue(v)</code>
<code> chroma_format_idc[i]</code>	<code>u(2)</code>
<code> pic_cropping_flag[i]</code>	<code>u(1)</code>
<code> if(pic_cropping_flag[i]) {</code>	
<code> pic_crop_left_offset[i]</code>	<code>ue(v)</code>
<code> pic_crop_right_offsetv[i]</code>	<code>ue(v)</code>
<code> pic_crop_top_offset[i]</code>	<code>ue(v)</code>
<code> pic_crop_bottom_offset[i]</code>	<code>ue(v)</code>
<code> }</code>	
<code> }</code>	
<code> num_layers_minus1</code>	<code>ue(v)</code>
<code> dependency_id_flag</code>	<code>u(1)</code>
<code> quality_id_flag</code>	<code>u(1)</code>
<code> view_id_flag</code>	<code>u(1)</code>
<code> region_id_flag</code>	<code>u(1)</code>
<code> layer_dependency_info-flag</code>	<code>u(1)</code>
<code> for (i = 1; i <= num_layers_minus1; i++) {</code>	

frame_size_idx[i]	ue(v)
rep_format_idx[i]	ue(v)
if(dependency_id_flag) {	
one_dependency_id_flag[i]	u(1)
if (one_dependency_id_flag[i] = = 1)	
dependency_id[i]	ue(v)
else if(one_dependency_id_flag[i] = = 0) {	
dependency_id_min[i]	ue(v)
dependency_id_max[i]	ue(v)
}	

Como se muestra en la Tabla 16, pic_width_in_luma_samples[i] y pic_height_in_luma_samples[i], bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y chroma_format_idc[i] pueden señalizarse a través de información sobre diferentes formatos de representación.

- 5 De acuerdo con otra realización, pic_width_in_luma_samples[i], pic_height_in_luma_samples[i], bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y chroma_format_idc[i] pueden señalizarse a través de información sobre imágenes diferentes, es decir imágenes que tienen resoluciones diferentes.

Una sintaxis para un mensaje de SEI de activación para la señalización de una velocidad de transmisión de bits e información de calidad es igual que en la Tabla 17.

10 Tabla 17

	Descriptor
bit rate_quality_info(payloadSize){	
num_layers_minus1	ue(v)
bit_rate_info_flag	u(1)
quality_info_flag	u(1)
quality_type_flag	u(1)
for(i = 1; i <= num_layers_minus1; i++) {	
if(bit_rate_info_flag) {	
max_bit rate[i]	u(8)
average_bit rate[i]	u(8)
}	
if(quality_info_flag) {	

quality_value[i]	u(8)
}	
}	
if(quality_type_flag) {	
QualityTypeUrildx = 0	
do	
quality_type_uri[QualityTypeUrildx]	b(8)
while(quality_type_uri[QualityTypeUrildx++] != 0)	
}	
}	

La semántica de la sintaxis de la Tabla 17 es como se indica a continuación.

- num_layers_minus1 indica el número de capas escalables que pueden proporcionarse en un tren de bits.

5 - Cuando el valor de bit_rate_info_flag es 1, indica que se proporciona información de la velocidad de transmisión de bits para capa escalable.

- Cuando el valor de quality_info_flag es 1, indica que se proporciona información de un valor de calidad de cada capa escalable.

- Cuando el valor de quality_type_flag es 1, indica que se proporciona información de un tipo de calidad para cada capa escalable.

10 - max_bit_rate[i] indica una velocidad máxima de transmisión de bits de una capa escalable cuyo valor layer_id es i, y average_bit_rate[i] indica una velocidad media de transmisión de bits de una capa escalable cuyo valor layer_id es i.

- quality_value[i] indica un valor de calidad de una capa escalable i.

15 - quality_type_uri[QualityTypeUrildx] indica un identificador universal de recursos (URI, por sus siglas en inglés) que tiene QualityTypeUrildx bytes de una cadena terminada en 0 nulo codificada en caracteres UTF-8 y que indica la inclusión de una expresión para el tipo de valores de calidad.

A continuación se propone una idea para mejorar un conjunto de parámetros de vídeo (VPS) con el fin de extraer eficazmente un tren de bits.

Referenciación de capas

20 Un método que indique una relación entre layer_id y un ID de dimensión de escalabilidad en un tren de datos que soporte múltiples capas puede incluir el primer método y el segundo método. El primer método informa de un método de mapeo entre layer_id y el ID de dimensión de escalabilidad. El segundo método parte o une los bits de layer_id y luego informa de qué tipo de dimensión está presente en el bit partido o unido.

25 En un tren de bits que soporte múltiples capas, un tipo de dimensión puede significar el tipo de escalabilidad, tal como la escalabilidad espacial y la escalabilidad de calidad, y un ID de dimensión puede significar un índice de una capa para un tipo de dimensión específico.

En un tren de bits que soporte múltiples capas, en general, una capa específica (para ayudar a entenderlo, por ejemplo en el caso en que se soporte la escalabilidad temporal en un tren de bits de una sola capa, una capa temporal (subcapa) 3) puede referirse directamente a la siguiente capa inferior (por ejemplo una capa temporal (subcapa)) en una dimensión específica.

Por ejemplo, en el caso de que se soporte la escalabilidad espacial, significa que una capa espacial 2 se refiere directamente a la siguiente capa espacial inferior 1.

Por consiguiente, para indicar el caso anterior, se propone que se describa en primer lugar una dimensión que tenga una dependencia directa por defecto.

- 5 Después, se describe detalladamente una dependencia específica en un bucle de descripción para una capa escalable.

A continuación se propone una idea para señalar una referenciación de capas utilizando los dos métodos. Una sintaxis mejorada para vps_extension es igual que en las Tablas 18 a 21.

Tabla 18

	Descriptor
vps_extension() {	
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bits	u(1)
all_default_dependency_flag	u(1)
if(all_default_dependency_flag == 0) {	
num_default_dim_minus1	u(4)
for(j = 0; j <= num_default_dim_minus1; j++) {	
dimension[j]	u(4)
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
num_dimensions_minus1[i]	u(4)
for (j = 0; j <= num_dimensions_minus1; j++) {	
dimension_type[i][j]	u(4)
dimension_id[i][j]	u(8)
}	
// layer dependency	
specific_dependency_flag[i]	u(1)
if(specific_dependency_flag[i] == 1) {	
num_direct_ref_layers[i]	u(6)
for (j = 0; j < num_direct_ref_layers[i]; j++)	

ref_layer_id[i][j]	u(6)
}	
}	
}	

La Tabla 18 muestra una sintaxis que mapea layer_id a un ID de dimensión de escalabilidad utilizando el primer método. La semántica de la sintaxis de la Tabla 18 es como se indica a continuación.

5 - Cuando el valor de all_default_dependency_flag es 1, indica que todas las dimensiones de capa tienen dependencia por defecto. Es decir, significa que, en una dimensión específica i, una capa que tiene 'dimension_id[i] = n' se refiere directamente a otra capa que tiene dimension_id[i] = n-1 por defecto.

Cuando el valor de all_default_dependency_flag es 0, indica que todas las dimensiones de capa pueden no tener dependencia por defecto. Cuando el valor de all_default_dependency_flag es 0, se señala el 'num_default_dim_minus1' siguiente.

10 - num_default_dim_minus1 indica el número de dimensiones que tienen dependencia por defecto.

- dimension[j] especifica el tipo de una dimensión de capa que tiene dependencia por defecto. Es decir que se señala información sobre el tipo de dimensión de capa que tiene dependencia por defecto, mientras se aumenta el número de dimensiones que tienen dependencia por defecto una a una. En una dimensión correspondiente, una capa superior (por ejemplo dimension_id = n) se referirá directamente a la siguiente capa inferior (por ejemplo dimension_id = n-1).

15 - Cuando el valor de specific_dependency_flag[i] es 1, significa que hay referencias/dependencias directas descritas detalladamente para una capa correspondiente. Por consiguiente, cuando el valor de specific_dependency_flag[i] es 1, se señalan el número de capas y el ID de las capas a las que se refiere directamente una capa correspondiente.

20 Que una capa C se refiera directamente a una capa B significa que un decodificador necesita utilizar información (decodificada o no decodificada) de la capa B para decodificar la capa C. Si la capa B utiliza directamente información de una capa A, no se considera que la capa C se refiera directamente a la capa A.

Tabla 19

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bits	u(1)
all-default-dependency_flag	u(1)
if(all_default_dependency_flag == 0) {	
num_default_dim_minus1	u(4)
for (j = 0; j <= num_default_dim_minus1; j++) {	
dimension[j]	u(4)
}	
}	
// scalability type and layer_id partitioning method	

num_dimensions_minus1	
for (i = 0; i <= num_dimensions_minus1; i++) {	
dimension_type[i]	u(4)
layer_id_dim_len[i]	u(3)
}	
for (i = 0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
specific_dependency_flag[i]	u(1)
if(specific_dependency_flag[i] = =1) {	
num_direct_ref_layers[i]	u(6)
for (j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	

La Tabla 19 muestra una sintaxis en la que los bits de layer_id están asignados a un tipo de dimensión de escalabilidad y la longitud de un tipo de dimensión asignado se señala utilizando el segundo método.

- 5 num_dimensions_minus1 descrito en la Tabla 19 indica el número de dimensiones de capa que están presentes en un encabezamiento de NALU. Es decir que se comprueba el número de dimensiones de capa presentes en el encabezamiento de NALU y se comprueban un tipo de capa presente en cada dimensión de capa correspondiente y el número de bits asignados al tipo de dimensión.

La sintaxis 'all_default_dependency_flag, num_default_dim_minus1, dimension[j] y specific_dependency_flag[i]' para la referenciación de capas descrita en la Tabla 19 tiene la misma semántica que la sintaxis descrita en la Tabla 18.

- 10 Las Tablas 20 y 21 describen sintaxis alternativas a las Tablas 18 y 19. La Tabla 20 muestra una sintaxis alternativa que indica dependencia por defecto cuando se utiliza el primer método, y la Tabla 21 muestra una sintaxis alternativa que indica dependencia por defecto cuando se utiliza el segundo método.

Tabla 20

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bits	u(1)

num_dimensions_minus1	u(4)
for (i = 0; i <= num_dimensions_minus1; i++) {	
dimension_type[i]	u(4)
default_dependency_flag[i]	u(1)
}	
for (i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
for (j = 0; j <= num_dimensions_minus1; j++) {	
dimension_id[i][j]	u(8)
}	
// layer dependency	
specific_dependency_flag[i]	u(1)
if(specific_dependency_flag[i] = =1) {	
num_direct_ref_layers[i]	u(6)
for (j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	

Tabla 21

vps_extension() {	Descriptor
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bits	u(1)
// scalability type and layer_id partitioning method	
num_dimensions_minus1	
for (i = 0; i <= num_dimensions_minus1; i++) {	

dimension_type[i]	u(4)
layer_id_dim_len[i]	u(3)
default_dependency_flag[i]	u(1)
}	
for (i = 0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
specific_dependency_flag[i]	u(1)
if(specific_dependency_flag[i] == 1) {	
num_direct_ref_layers[i]	u(6)
for (j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	

Entre las sintaxis de las Tablas 20 y 21 se omite la descripción de las sintaxis descritas en las Tablas 18 y 19.

5 Una nueva sintaxis 'default_dependency_flag[i]' en las Tablas 20 y 21 indica si un tipo i de dimensión utiliza o no dependencia por defecto. En una dimensión correspondiente, una capa superior (por ejemplo dimension_id[i]=n) se refiere directamente a una capa situada justo debajo (por ejemplo dimension_id[i]=n-1).

Es decir que después de que num_dimensions_minus1 y dimension_type[i] designen un tipo de dimensión específico, se señala si el tipo de dimensión específico utiliza o no dependencia por defecto. Si no se señala, esto indica que se señala información de una capa a la que se refiere directamente la capa correspondiente.

En la Tabla 22 figuran tipos de dimensión según la presente invención.

10 Tabla 22

dimension_type[i][j]	dimension_id[i][j]
0	view order idx
1	depth flag
2	dependency ID
3	quality ID
4	priority ID

5	region ID
6..15	reservado

De acuerdo con la presente invención, los tipos de dimensión 4 y 5, es decir los tipos indicativos de un ID de prioridad y un ID de zona, se han añadido en un tipo de dimensión existente.

5 dimension_type[i][j] puede tener un valor entre 0 y 5. Otros valores pueden definirse posteriormente, y un decodificador puede ignorar un valor de dimension_type[i][j] si dimension_type[i][j] no tiene un valor entre 0 y 5.

Si dimension_type tiene un valor de 4, el dimension_id correspondiente indica el ID de una capa de prioridad de un tren de bits en el estándar SVC.

Si dimension_type tiene un valor de 5, el dimension_id correspondiente indica el ID de una zona específica de un tren de bits. La zona específica puede ser uno o más segmentos espacio-temporales en el tren de bits.

10 La Figura 4 es un diagrama de flujo de control que ilustra un método de codificación de información de vídeo de acuerdo con la presente invención.

Remitiéndonos a la Figura 4, el aparato de codificación codifica una unidad de capa de abstracción de red (NAL) que incluye información relacionada con vídeo en la etapa S401.

15 El encabezamiento de NALU de la NALU no incluye información que indique si la NALU incluye o no una porción que incluya al menos parte o la totalidad de la imagen no de referencia.

Entretanto, el encabezamiento de NALU incluye información de ID de capa para identificar una capa escalable en un tren de datos que soporte una capa escalable.

20 Aquí puede utilizarse para señalar la información de ID de capa un bit utilizado para señalar información que indique si una NALU, excepto el encabezamiento de NALU, incluye o no una porción que incluya al menos parte o la totalidad de la imagen no de referencia.

Además, la NALU puede incluir información sobre diversos conjuntos de parámetros necesarios para decodificar vídeo.

El aparato de codificación puede codificar un mensaje de información de mejora suplementaria (SEI), que incluya información sobre conjuntos de parámetros activos, como una NALU independiente.

25 La información sobre los conjuntos de parámetros activos puede incluir información sobre qué VPS activo está indexado y/o información sobre qué SPS activo está indexado.

Además, la información sobre conjuntos de parámetros activos puede incluir información sobre qué VPS activo está indexado, información sobre el número de SPS que se refieren al VPS activo e información sobre que SPS están indexados.

30 El aparato de decodificación puede utilizar la información sobre los conjuntos de parámetros para extraer una subcapa que proporcione escalabilidad temporal.

Además, el aparato de decodificación o un módulo de decodificación para realizar una decodificación puede utilizar la información sobre los conjuntos de parámetros a la hora de activar conjuntos de parámetros necesarios para decodificar una NALU de capa de codificación de vídeo.

35 El aparato de codificación envía la NALU, que incluye la información relacionada con el vídeo codificado, en forma de un tren de bits en la etapa S402.

La Figura 5 es un diagrama de flujo de control que ilustra un método de decodificación de información de vídeo de acuerdo con la presente invención

Remitiéndonos a la Figura 5, el aparato de decodificación recibe una NALU, que incluye información relacionada con un vídeo codificado, a través de un tren de bits en la etapa S501.

40 El aparato de decodificación analiza sintácticamente el encabezamiento y la carga útil de NAL de la NALU en la etapa S502. El análisis sintáctico de la información de vídeo puede realizarse mediante un módulo de decodificación entrópica o un módulo de análisis sintáctico adicional.

A través del análisis sintáctico, el aparato de decodificación puede obtener diversos datos incluidos en el encabezamiento y la carga útil de NAL de la NALU.

El encabezamiento de NALU puede incluir información de ID de capa para identificar una capa escalable en el tren de bits que soporta la capa escalable y puede no incluir información de etiqueta de 1 bit que indique si la NALU es una imagen no de referencia o una imagen de referencia en la totalidad del tren de bits a la hora de codificar los datos de vídeo.

- 5 Aquí puede utilizarse para señalar la información de ID de capa un bit utilizado para señalar información que indique si una NALU, excepto el encabezamiento de NALU, incluye o no una porción que incluya al menos parte o la totalidad de la imagen no de referencia.

- 10 Además, a través del análisis sintáctico, el aparato de decodificación puede obtener información sobre conjuntos de parámetros incluida en un mensaje de SEI. La información de los conjuntos de parámetros obtenida es necesaria para decodificar una NALU asociada con un mensaje de SEI.

La información sobre conjuntos de parámetros activos puede incluir información sobre qué VPS activo está indexado y/o información sobre qué SPS activo está indexado.

- 15 Además, la información sobre los conjuntos de parámetros activos puede incluir información sobre qué VPS activo está indexado, información indicativa del número de SPS que se refieren al VPS activo e información sobre que SPS están indexados.

El aparato de decodificación puede utilizar estos datos sobre conjuntos de parámetros para extraer una subcapa que proporcione escalabilidad temporal.

- 20 Además, los datos sobre conjuntos de parámetros pueden utilizarse a la hora de decodificar un tren de bits o en una negociación de sesión (por ejemplo una negociación de sesión en el momento de una transmisión en continuo (*streaming*) en una red IP).

- 25 En las realizaciones anteriormente mencionadas, aunque los métodos se han descrito sobre la base de los diagramas de flujo en forma de una serie de etapas o bloques, la presente invención no está limitada a la secuencia de las etapas, y algunas de las etapas pueden realizarse en un orden diferente del de otras etapas o pueden realizarse simultáneamente a otras etapas. Además, los expertos en la técnica entenderán que las etapas mostradas en el diagrama de flujo no son exclusivas y las etapas pueden incluir etapas adicionales o que una o más etapas del diagrama de flujo pueden borrarse sin afectar al alcance de la presente invención.

REIVINDICACIONES

1. Un aparato de decodificación de vídeo, que comprende:

un módulo de decodificación para decodificar una primera imagen;

5 un módulo de análisis sintáctico para analizar sintácticamente un encabezamiento de porción de una segunda imagen que es la imagen siguiente a la primera imagen decodificada; y

una memoria intermedia de imágenes decodificadas (DPB, por sus siglas en inglés) para almacenar la primera imagen decodificada,

en donde el encabezamiento de porción de la segunda imagen incluye información de imagen de referencia que indica si la primera imagen decodificada es una imagen de referencia,

10 en donde la primera imagen decodificada se marca como imagen de referencia en la DPB y luego se determina si la primera imagen decodificada marcada como imagen de referencia es una imagen de referencia o una imagen no de referencia sobre la base de la información de imagen de referencia, y la primera imagen decodificada marcada como imagen de referencia se marca como imagen no de referencia en caso de que se determine que la primera imagen decodificada es una imagen no de referencia sobre la base de la información de imagen de referencia incluida en el

15 encabezamiento de porción de la segunda imagen,

en donde el módulo de análisis sintáctico analiza sintácticamente un encabezamiento de unidad de capa de abstracción de red (NAL, por sus siglas en inglés) de una unidad de NAL para la primera imagen, y el encabezamiento de unidad de NAL no comprende información de etiqueta que indique si la unidad de NAL es una imagen no de referencia o una imagen de referencia.

FIG. 1

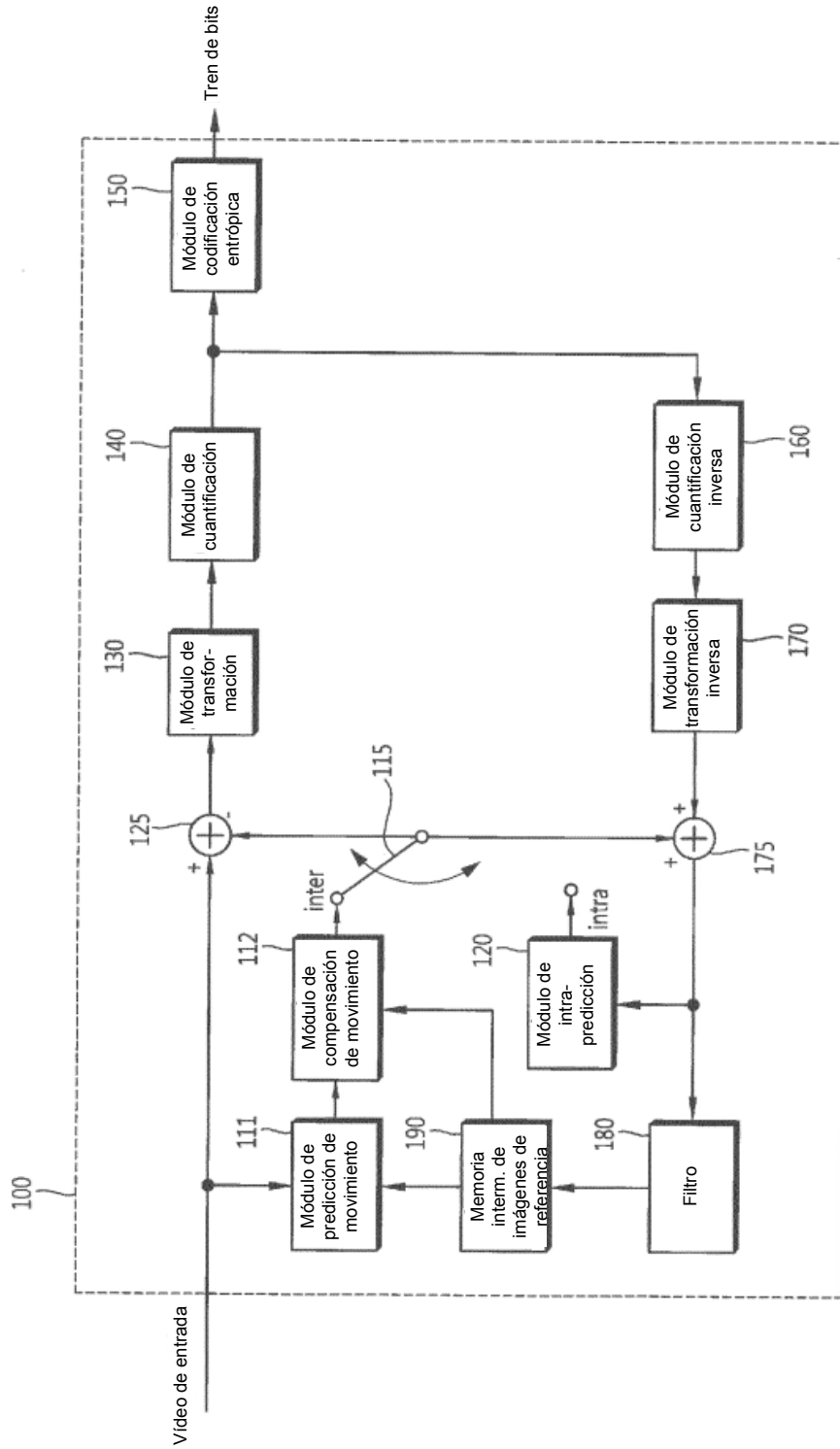


FIG. 2

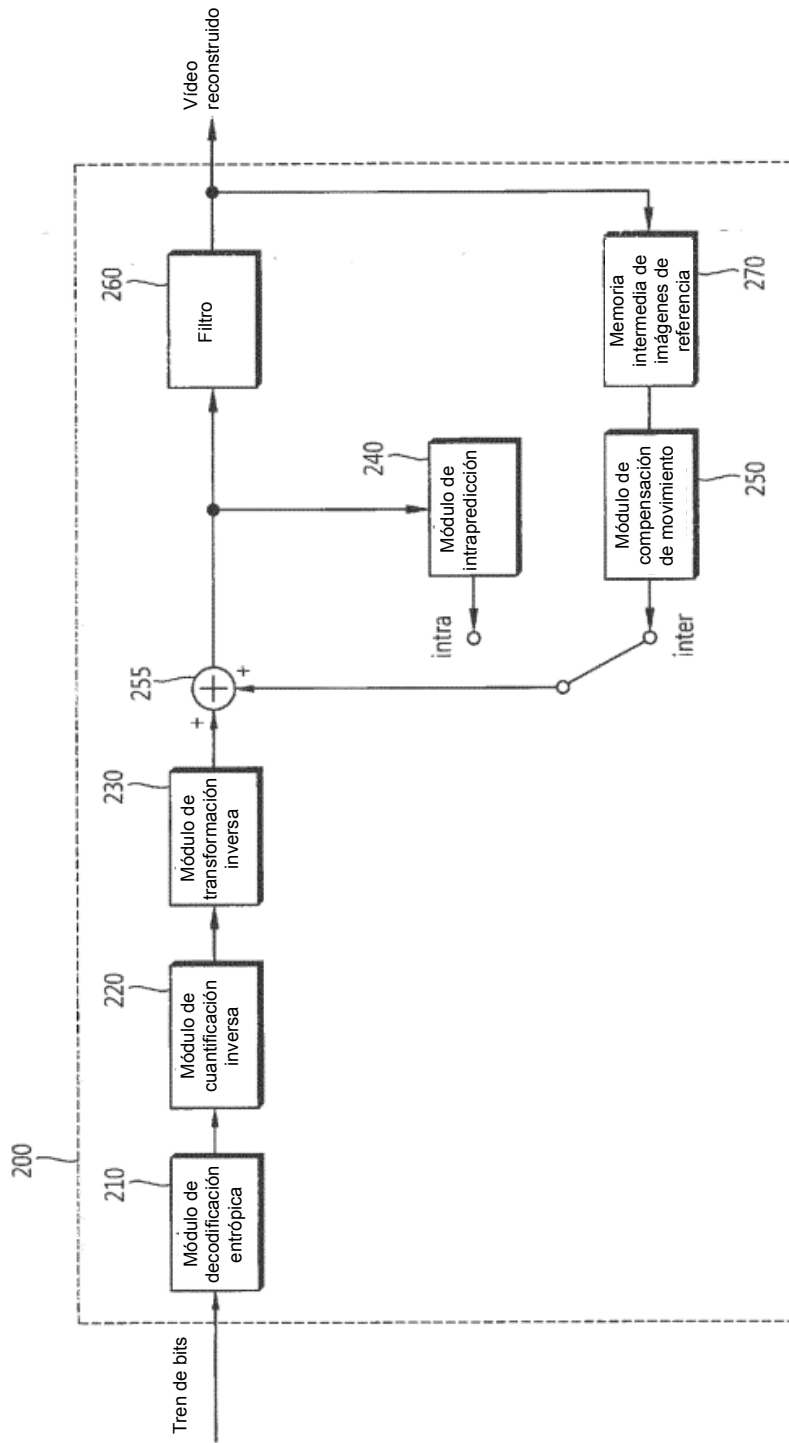


FIG. 3

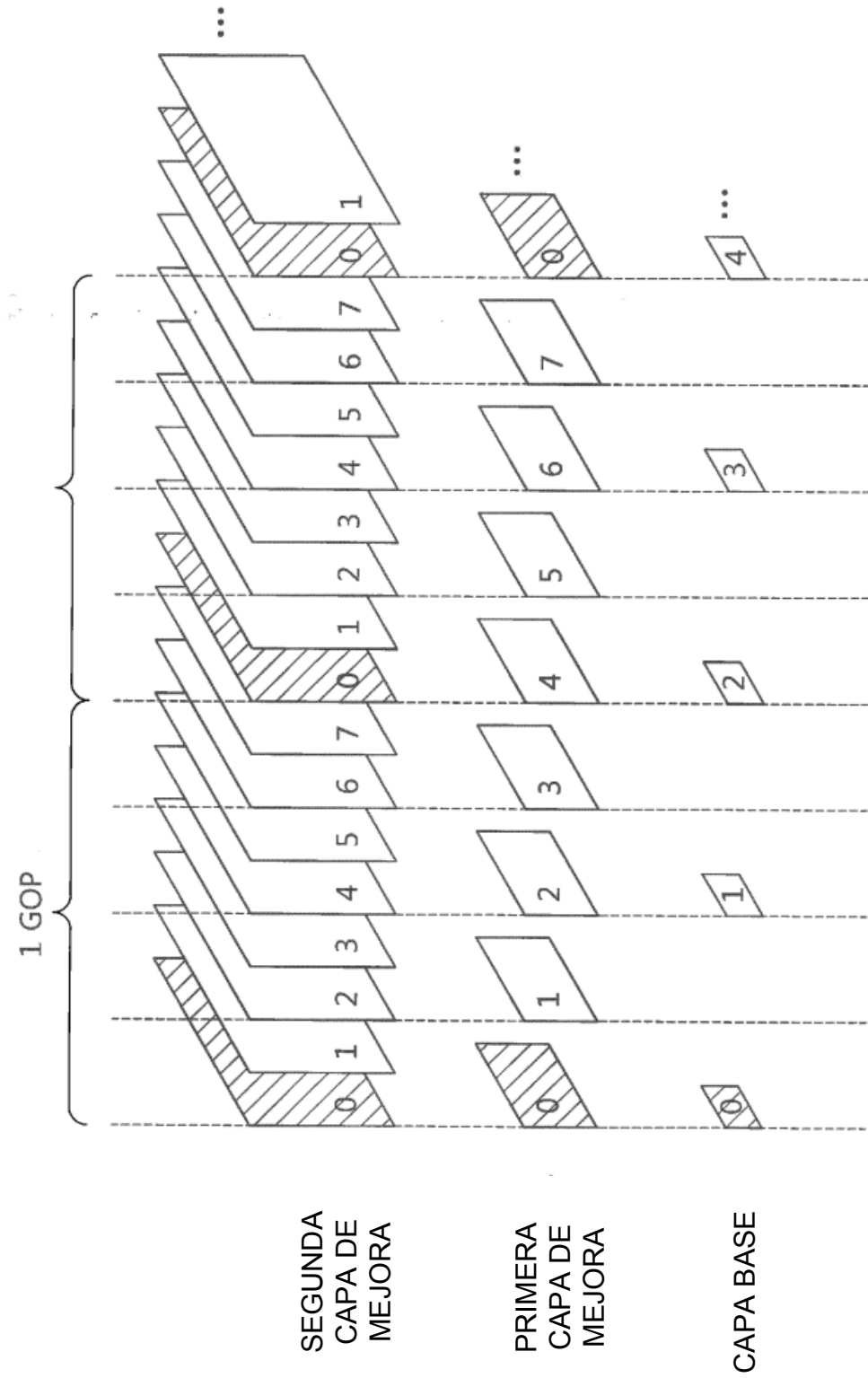


FIG. 4



FIG. 5

