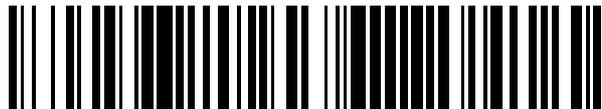


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 649 757**

51 Int. Cl.:

H04N 19/597 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.12.2007 PCT/US2007/024986**

87 Fecha y número de publicación internacional: **24.07.2008 WO08088497**

96 Fecha de presentación y número de la solicitud europea: **06.12.2007 E 07862581 (1)**

97 Fecha y número de publicación de la concesión europea: **01.11.2017 EP 2103136**

54 Título: **Métodos y aparatos para señalización mejorada que utilizan sintaxis de alto nivel para la codificación y descodificación de vídeo de múltiples vistas**

30 Prioridad:

21.12.2006 US 871401 P

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

15.01.2018

73 Titular/es:

**THOMSON LICENSING (100.0%)
1-5, rue Jeanne d'Arc
92130 Issy-les-Moulineaux, FR**

72 Inventor/es:

**PANDIT, PURVIN BIBHAS;
SU, YEPING y
YIN, PENG**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 649 757 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

Métodos y aparatos para señalización mejorada que utilizan sintaxis de alto nivel para la codificación y descodificación de vídeo de múltiples vistas

Campo técnico

5 Los presentes principios se refieren en general a la codificación y descodificación de video y, más particularmente, a métodos y aparatos para señalización mejorada que utilizan sintaxis de alto nivel para la codificación y descodificación de video de múltiples vistas.

Antecedentes

10 En la versión actual de la extensión de la codificación de video de múltiples vistas (MVC – Multi-view Video Coding, en inglés) basada en el estándar Advanced Video Coding (AVC), Parte 10, de la International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group-4 (MPEG-4) / Recomendación H.264 de la International Telecommunication Union – Telecommunication Sector (ITU-T) (a continuación en este documento, " estándar AVC del MPEG-4 "), se agrega una nueva sintaxis en la extensión de MVC del conjunto de parámetros de secuencia para señalar las referencias entre vistas. Es decir, esta sintaxis se utiliza para indicar las referencias entre vistas que se utilizarán para las imágenes vínculo y las que no lo son.

15 El documento VETRO A. ET AL.: "Joint Multiview Video Model (JMVM) 1.0 ", JOINT VIDEO TEAM (JVT) OF ISO/IEC MPEG & ITU-T VCEG, [En línea] 16 de agosto de 2006 (2006-08-16), Klagenfurt, Austria da a conocer esta sintaxis de MVC.

20 Esta sintaxis puede servir asimismo para construir el mapa de dependencia de vista, que puede ser utilizado para acceso aleatorio a vista, un proceso de marcado de imagen de referencia descodificada implícita, y otros. La sintaxis se define mediante la realización de un bucle de la variable view_id de 0 al número de vistas menos 1. view_id indica el identificador de vista para una vista. No incluye ninguna información de orden de codificación.

25 Esta definición establece algunas restricciones sobre cómo asignar una view_id a cada vista. La definición no permite espacios en los view_id. Si más tarde, algunas vistas no están codificadas, la definición causará cierta ambigüedad. Además, esta definición carece de algunas características. Por ejemplo, la definición no especifica el orden de codificación de cada vista.

La TABLA 1 ilustra la sintaxis de la extensión de la codificación de video múltiples vistas (MVC) del conjunto de parámetros de secuencia (SPS – Sequence Parameter Set, en inglés).

TABLA 1

	C	Descriptor
seq_parameter_set_mvc_extension() {		
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		
}		

30

La semántica de las sintaxis en la TABLA 1 se define de la siguiente manera:

num_views_minus_1 más 1 identifica el número total de vistas en el flujo de bits. El valor de **number_of_view_minus_1** estará en el rango de 0 a 1023.

5 **num_anchor_refs_I0** [i] especifica el número de referencias de predicción entre vistas para list0 para imágenes de vínculo con **view_id** igual a i. El valor de **num_anchor_refs_I0** [i] será menor o igual que **num_ref_frames**.

anchor_ref_I0 [i] [j] identifica el **view_id** de la vista que se utiliza como la referencia de orden j de list0, para las imágenes de vínculo de la vista con **view_id** igual a i.

num_anchor_refs_I1 [i] especifica el número de referencias de predicción entre vistas para list1 de la imagen de vínculo con **view_id** igual a i. El valor de **num_anchor_refs_I1** [i] será menor o igual que **num_ref_frames**.

10 **anchor_ref_I1** [i] [j] identifica el **view_id** de la vista que se utiliza como la referencia de orden j de list1, para las imágenes de vínculo de la vista con **view_id** igual a i.

num_non_anchor_refs_I0 [i] especifica el número de referencias de predicción entre vistas para list0 para imágenes no de vínculo con **view_id** igual a i. El valor de **num_anchor_refs_I0** [i] será menor o igual que **num_ref_frames**.

15 **non_anchor_ref_I0** [i] [j] identifica el **view_id** de la vista que se utiliza como la referencia de orden j de list0, para las imágenes no de vínculo de la vista con **view_id** igual a i.

num_non_anchor_refs_I1 [i] especifica el número de referencias de predicción entre vistas para list1 para imágenes no de vínculo con **view_id** igual a i. El valor de **num_anchor_refs_I1** [i] será menor o igual que **num_ref_frames**.

20 **non_anchor_ref_I1** [i] [j] identifica el **view_id** de la vista que se utiliza como la referencia de orden j de list1, para las imágenes no de vínculo de la vista con **view_id** igual a i.

El siguiente procedimiento se realizará para colocar imágenes de referencia de una vista que es diferente de la vista actual en las listas de predicción de referencia. Si la imagen actual es una imagen de vínculo, entonces para cada valor de i de 0 a **num_anchor_refs_IX** - 1, la imagen con **view_id** igual a **anchor_ref_IX** [i] y con el mismo **PicOrderCnt** () que la imagen actual se añadirá a **RefPicListX**. De lo contrario, si la imagen actual no es una imagen de vínculo, entonces para cada valor de i de 0 a **num_non_anchor_refs_IX** - 1, la imagen con **view_id** igual a **non_anchor_ref_IX** [i] y con el mismo **PicOrderCnt** () que la imagen actual se añadirá a **RefPicListX**.

25 A partir de la TABLA 1, se puede ver que la sintaxis en el conjunto de parámetros de secuencia (SPS) de la extensión de codificación de video de múltiples vistas (MVC) se define realizando un bucle sobre la variable **view_id** desde 0 hasta el número de vistas menos 1. Esta definición establece cierta restricción sobre cómo asignar un **view_id** para cada vista. La definición no permite espacios en los **view_id**. Si más tarde, por algún motivo o motivos tal como la limitación del ancho de banda o el interés de un usuario, algunas vistas no están codificadas, esta definición causará cierta ambigüedad. Para mantener la continuidad de la asignación de **view_id**, se puede implementar lo siguiente.

35 En un primer método de la técnica anterior para mantener la continuidad de la asignación de **view_id**, se establece la información de vista no codificada en el conjunto de parámetros de secuencia y el número de referencias entre vistas se establece en 0. Sin embargo, esto no es deseable, ya que el descodificador no sabrá si la vista no codificada falta intencionalmente o debido a errores de transmisión.

40 En un segundo método de la técnica anterior para mantener la continuidad de la asignación de **view_id**, la información de la vista no codificada es eliminada del conjunto de parámetros de secuencia. Sin embargo, esto causará huecos de **view_id**. Por lo tanto, los **view_id** deben ser reasignados. Esto tampoco es deseable ya que **view_id** ya no es exclusivo para cada vista.

Compendio

45 Estos y otros inconvenientes y desventajas de la técnica anterior se abordan mediante los presentes principios, que están dirigidos a métodos y aparatos para mejorar la señalización utilizando sintaxis de alto nivel para codificación de video y descodificación de múltiples vistas.

50 De acuerdo con un aspecto de los presentes principios, se proporciona un aparato. El aparato incluye un codificador para codificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits resultante, en el que dicho codificador señala la información de orden de codificación de video para al menos la al menos una imagen en una sintaxis de alto nivel.

De acuerdo con otro aspecto de los presentes principios, se proporciona un método. El método incluye codificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits resultante, en el que dicha etapa de codificación comprende señalar la información de orden de codificación de video para al menos la al menos una imagen en una sintaxis de alto nivel.

- 5 De acuerdo con otro aspecto más de los presentes principios, se proporciona un aparato. El aparato incluye un descodificador para descodificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits resultante, en el que dicho descodificador determina la información de orden de codificación de video para al menos la al menos una imagen de una sintaxis de alto nivel.

- 10 De acuerdo con otro aspecto de los presentes principios, se proporciona un método. El método incluye descodificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits resultante, en el que dicha etapa de codificación comprende determinar la información de orden de codificación de video para al menos la al menos una imagen de una sintaxis de alto nivel.

- 15 Estos y otros aspectos, características y ventajas de los presentes principios resultarán evidentes a partir de la siguiente descripción detallada de realizaciones a modo de ejemplo, que debe ser leída junto con los dibujos adjuntos.

Breve descripción de los dibujos

Los presentes principios se pueden comprender mejor de acuerdo con las siguientes figuras a modo de ejemplo, en las que:

- 20 la figura 1 es un diagrama de bloques para un codificador de codificación de video de múltiples vistas (MVC) a modo de ejemplo al cual se pueden aplicar los presentes principios, de acuerdo con una realización de los presentes principios;

la figura 2 es un diagrama de bloques para un descodificador de codificación de video de múltiples vistas (MVC) a modo de ejemplo al que se pueden aplicar los presentes principios, de acuerdo con una realización de los presentes principios;

- 25 la figura 3 es un diagrama para una estructura de codificación de tiempo-primero para un sistema de codificación de video de múltiples vistas con 8 vistas al que se pueden aplicar los presentes principios, de acuerdo con una realización de los presentes principios;

la figura 4 es un diagrama de flujo para un método a modo de ejemplo para codificar el contenido de video de múltiples vistas, de acuerdo con una realización de los presentes principios;

- 30 la figura 5 es un diagrama de flujo para otro método a modo de ejemplo para codificar el contenido de video de múltiples vistas, de acuerdo con una realización de los presentes principios;

la figura 6 es un diagrama de flujo para un método a modo de ejemplo para descodificar el contenido de video de múltiples vistas, de acuerdo con una realización de los presentes principios;

- 35 la figura 7 es un diagrama de flujo para otro método a modo de ejemplo para descodificar el contenido de video de múltiples vistas, de acuerdo con una realización de los presentes principios;

Descripción detallada

Los presentes principios están dirigidos a métodos y aparatos para una señalización mejorada utilizando sintaxis de alto nivel para la codificación y descodificación de video de múltiples vistas.

- 40 La presente descripción ilustra los presentes principios. Por lo tanto, se apreciará que los expertos en la materia podrán diseñar diversas disposiciones que, aunque no describen o muestran explícitamente en este documento, incorporan los presentes principios y están incluidas en su espíritu y alcance,

- 45 Todos los ejemplos y el lenguaje condicional que se citan en este documento están destinados a fines pedagógicos, para ayudar al lector a comprender los presentes principios y los conceptos aportados por el inventor o los inventores para promover la técnica, y deben ser interpretados sin limitación a dichos ejemplos y condiciones citados específicamente.

- Además, todas las afirmaciones en este documento que citan principios, aspectos y realizaciones de los presentes principios, así como ejemplos específicos de los mismos, pretenden abarcar tanto equivalentes estructurales como funcionales de los mismos. Además, se pretende que dichos equivalentes incluyan tanto equivalentes actualmente conocidos como equivalentes desarrollados en el futuro, es decir, cualquier elemento desarrollado que realice la misma función, independientemente de la estructura.

De este modo, por ejemplo, los expertos en la técnica apreciarán que los diagramas de bloques presentados en este documento representan vistas conceptuales de circuitos ilustrativos que incorporan los presentes principios. De manera similar, se apreciará que cualquier gráfico de flujo, diagrama de flujo, diagrama de transición de estado, pseudocódigo y similares representan diversos procesos que se pueden representar sustancialmente en medios legibles por ordenador y pueden ser ejecutados por un ordenador o procesador, se muestre o no explícitamente dicho ordenador o procesador.

Las funciones de los diversos elementos que se muestran en las figuras pueden ser proporcionados mediante la utilización de hardware dedicado, así como de hardware capaz de ejecutar software en asociación con el software apropiado. Cuando son proporcionadas por un procesador, las funciones pueden ser proporcionadas por un único procesador dedicado, por un único procesador compartido, o por una pluralidad de procesadores individuales, algunos de los cuales pueden ser compartidos. Además, la utilización explícita del término "procesador" o "controlador" no debe ser interpretada como referida exclusivamente a hardware capaz de ejecutar software, y puede incluir implícitamente, sin limitación, hardware de procesador de señal digital ("DSP" – Digital Signal Processor, en inglés), de solo lectura. memoria ("ROM" – Read - Only Memory, en inglés) para almacenar software, memoria de acceso aleatorio ("RAM" – Random Access Memory, en inglés) y almacenamiento no volátil.

Se puede incluir asimismo otro hardware, convencional y/o personalizado. De manera similar, todos los conmutadores que se muestran en las figuras son solo conceptuales. Su función puede ser llevada a cabo mediante la operación de la lógica del programa, mediante lógica dedicada, mediante la interacción del control del programa y la lógica dedicada, o incluso manualmente, siendo la técnica particular seleccionable por el implementador tal como se entiende más específicamente a partir el contexto.

En las reivindicaciones de este documento, cualquier elemento expresado como un medio para realizar una función específica tiene por objeto abarcar cualquier forma de realizar esa función, incluyendo, por ejemplo, a) una combinación de elementos de circuito que realiza esa función o b) software en cualquier forma, incluyendo, por lo tanto, firmware, microcódigo o similar, combinado con circuitos apropiados para ejecutar ese software para realizar la función. Los presentes principios definidos por tales reivindicaciones residen en el hecho de que las funcionalidades proporcionadas por los diversos medios citados se combinan y se unen de la manera que requieren las reivindicaciones. Por lo tanto, se considera que cualquier medio que pueda proporcionar esas funcionalidades es equivalente a los mostrados en este documento.

La referencia en la memoria descriptiva a "una realización" de los presentes principios significa que una función, estructura, característica y otros, particular, descrita en relación con la realización está incluida en al menos una realización de los presentes principios. Por lo tanto, las apariciones de la frase "en una realización" que aparecen en varios lugares a lo largo de la memoria descriptiva no necesariamente se refieren a la misma realización.

Tal como se utiliza en este documento, "sintaxis de alto nivel" se refiere a la sintaxis presente en el flujo de bits que reside jerárquicamente por encima de la capa de macrobloque. Por ejemplo, la sintaxis de alto nivel, tal como se utiliza en este documento, puede referirse, pero no se limita a, sintaxis al nivel del encabezado del segmento, al nivel del conjunto de parámetros de secuencia (SPS), al nivel del conjunto de parámetros de imagen (PPS – Picture Parameter Set, en inglés), al nivel del conjunto de parámetros de vista (VPS – View Parameter Set, en inglés), al nivel del encabezado de la unidad de capa de abstracción de la red (NAL – Network Abstraction Layer, en inglés) y en un mensaje de información suplementaria de mejora (SEI – Supplemental Enhancement Information, en inglés).

Por motivos de ilustración y brevedad, las siguientes realizaciones se describen en este documento con respecto a la utilización de una sintaxis de alto nivel en el conjunto de parámetros de secuencia. Sin embargo, se debe apreciar que los presentes principios no están limitados únicamente a la utilización del conjunto de parámetros de secuencia con respecto a la señalización mejorada dada a conocer en este documento y, por lo tanto, dicha señalización mejorada puede ser implementada con respecto a al menos los tipos descritos anteriormente de sintaxis de alto nivel que incluyen, pero no están limitados a, sintaxis al nivel del encabezado del segmento, al nivel del conjunto de parámetros de secuencia (SPS), al nivel del conjunto de parámetros de imagen (PPS), al nivel del conjunto de parámetros de vista (VPS), al nivel del encabezado de la unidad de capa de abstracción de la red (NAL), y en un mensaje de información suplementaria de mejora (SEI), manteniendo al mismo tiempo el espíritu de los presentes principios.

Se apreciará adicionalmente que aunque una o más realizaciones de los presentes principios se describen en este documento con respecto al estándar de extensión de múltiples vistas de la AVC de MPEG-4, los presentes principios no están limitados únicamente a este estándar y, por lo tanto, pueden ser utilizados con respecto a otros estándares de codificación de video, recomendaciones y extensiones de los mismos, incluyendo extensiones del estándar de extensión de múltiples vistas de la AVC de MPEG-4, manteniendo el espíritu de los presentes principios.

Además, se debe apreciar que la utilización del término "y/o", por ejemplo, en el caso de "A y/o B", pretende abarcar la selección de la primera opción enumerada (A), la selección de la segunda opción enumerada (B) o la selección de ambas opciones (A y B). Como ejemplo adicional, en el caso de "A, B y/o C", dichas frases pretenden abarcar la selección de la primera opción enumerada (A), la selección de la segunda opción enumerada (B), la selección de la tercera opción enumerada (C), la selección de las primera y segunda opciones enumeradas (A y B), la selección de

las primera y tercera opciones enumeradas (A y C), la selección de las segunda y tercera opciones enumeradas (B y C), o la selección de las tres opciones (A y B y C). Esto se puede extender, como resulta fácilmente evidente para un experto en esta técnica y en las relacionadas, a cualquier número de elementos enumerados.

5 Volviendo a la figura 1, un codificador de codificación de video de múltiples vistas (MVC) a modo de ejemplo está indicado en general por el número de referencia 100. El codificador 100 incluye un combinador 105 que tiene una salida conectada en comunicación de señal con una entrada de un transformador 110. Una salida del transformador 110 está conectada en comunicación de señal con una entrada del cuantificador 115. Una salida del cuantificador 115 está conectada en comunicación de señal con una entrada de un codificador de entropía 120 y una entrada de un cuantificador inverso 125. Una salida del cuantificador inverso 125 está conectada en comunicación de señal con una entrada de un transformador inverso 130. Una salida del transformador inverso 130 está conectada en comunicación de señal con una primera entrada no inversora de un combinador 135. Una salida del combinador 135 está conectada en comunicación de señal con una entrada de un intra predictor 145 y una entrada de un filtro de desbloqueo 150. Una salida del filtro de desbloqueo 150 está conectada en comunicación de señal con una entrada de un almacén de imágenes de referencia 155 (para la vista i). Una salida del almacén de imágenes de referencia 155 está conectada en comunicación de señal con una primera entrada de un compensador de movimiento 175 y una primera entrada de un estimador de movimiento 180. Una salida del estimador de movimiento 180 está conectada en comunicación de señal con una segunda entrada del compensador de movimiento 175.

20 Una salida de un almacén de imágenes de referencia 160 (para otras vistas) está conectada en comunicación de señal con una primera entrada de un estimador de disparidad 170 y una primera entrada de un compensador de disparidad 165. Una salida del estimador de disparidad 170 está conectada en comunicación de señal con una segunda entrada del compensador de disparidad 165.

25 Una salida del descodificador de entropía 120 está disponible como una salida del codificador 100. Una entrada no inversora del combinador 105 está disponible como una entrada del codificador 100, y está conectada en comunicación de señal con una segunda entrada del estimador de disparidad 170, y una segunda entrada del estimador de movimiento 180. Una salida de un conmutador 185 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 135 y con una entrada inversora del combinador 105. El conmutador 185 incluye una primera entrada conectada en comunicación de señal con una salida del compensador de movimiento 175, una segunda entrada conectada en comunicación de señal con una salida del compensador de disparidad 165, y una tercera entrada conectada en comunicación de señal con una salida del intra predictor 145.

30 Volviendo a la figura 2, un descodificador de codificación de video de múltiples vistas (MVC) a modo de ejemplo se indica en general con el número de referencia 200. El descodificador 200 incluye un descodificador de entropía 205 que tiene una salida conectada en comunicación de señal con una entrada de un cuantificador inverso 210. Una salida del cuantificador inverso está conectada en comunicación de señal con una entrada de un transformador inverso 215. Una salida del transformador inverso 215 está conectada en comunicación de señal con una primera entrada no inversora de un combinador 220. Una salida del combinador 220 está conectada en comunicación de señal con una entrada de un filtro de desbloqueo 225 y una entrada de un intra predictor 230. Una salida del filtro de desbloqueo 225 está conectada en comunicación de señal con una entrada de un almacén de imágenes de referencia 240 (para la vista i). Una salida del almacén de imágenes de referencia 240 está conectada en comunicación de señal con una primera entrada de un compensador de movimiento 235.

40 Una salida de un almacén de imágenes de referencia 245 (para otras vistas) está conectada en comunicación de señal con una primera entrada de un compensador de disparidad 250.

45 Una entrada del codificador de entropía 205 está disponible como una entrada al descodificador 200, para recibir un flujo de bits de residuo. Además, una entrada de control del conmutador 255 está asimismo disponible como una entrada al descodificador 200, para recibir la sintaxis de control para controlar qué entrada es seleccionada mediante el conmutador 255. Además, una segunda entrada del compensador de movimiento 235 está disponible como una entrada del descodificador 200, para recibir vectores de movimiento. Además, una segunda entrada del compensador de disparidad 250 está disponible como una entrada al descodificador 200, para recibir vectores de disparidad.

50 Una salida de un conmutador 255 está conectada en comunicación de señal con una segunda entrada no inversora del combinador 220. Una primera entrada del conmutador 255 está conectada en comunicación de señal con una salida del compensador de disparidad 250. Una segunda entrada del conmutador 255 está conectada en comunicación de señal con una salida del compensador de movimiento 235. Una tercera entrada del conmutador 255 está conectada en comunicación de señal con una salida del intra predictor 230. Una salida del módulo de modo 260 está conectada en comunicación de señal con el conmutador 255 para controlar qué entrada es seleccionada mediante el conmutador 255. Una salida del filtro de desbloqueo 225 está disponible como una salida del descodificador.

De acuerdo con los presentes principios, se proporcionan métodos y aparatos para mejorar la señalización utilizando una sintaxis de alto nivel para codificación y descodificación de video de múltiples vistas.

Tal como se ha indicado anteriormente, la extensión de la codificación actual de video de múltiples vistas del estándar AVC de MPEG-4 incluye una sintaxis que señala las referencias entre vistas. Sin embargo, tal como se señaló anteriormente, la definición de sintaxis carece de ciertas características. Por ejemplo, la definición de sintaxis no especifica el orden de codificación de cada vista. Esta información es muy útil para la recuperación de errores. Si se define correctamente, puede ayudar al descodificador a detectar la falta de algunas imágenes en ciertas vistas y el descodificador podría ocultar la imagen perdida en ciertas vistas sin perder la pista de lo que estaba sucediendo.

Volviendo a la figura 3, una estructura de codificación primero en el tiempo (Time-First, en inglés) para un sistema de codificación de video de múltiples vistas con 8 vistas se indica en general mediante el número de referencia 300. En el ejemplo de la figura 3, todas las imágenes en el mismo instante de tiempo desde diferentes puntos de vista se codifican de manera contigua. Por lo tanto, si conocemos la información del orden de codificación, entonces podemos detectar rápidamente qué vista se pierde en un instante al rastrear el `view_id`. Esta información puede ayudar asimismo a acelerar la construcción del mapa de dependencia. Por ejemplo, vistas codificadas anteriores no utilizarán vistas codificadas posteriores como referencias. El mapa de dependencia puede ser utilizado para, por ejemplo, acceso aleatorio, un proceso de marcado implícito de la imagen de referencia descodificada, y así sucesivamente.

Para construir la información del mapa de dependencia a partir de la información de dependencia de la vista actual en el conjunto de parámetros de secuencia, se puede utilizar una llamada recursiva. En el siguiente ejemplo, la indicación de una imagen de vínculo / no de vínculo se elimina porque el algoritmo es aplicable a ambos.

Proponemos la construcción de `picture_dependency_maps [i] [j]` con la siguiente semántica:

`picture_dependency_maps [i] [j]` igual a 1 indica que la imagen con `view_id` igual a `j` dependerá de la imagen con `view_id` igual a `i`.

Se supone que estamos utilizando estructuras estáticas de datos. Es deseable obtener la siguiente información para construir los `picture_dependency_maps`:

- `num_refs [k]`, indicando el número de referencias entre vistas para la vista con `view_id` igual a `k`
- la lista de `view_id` a los que hace referencia la vista con `view_id` igual a `k`: `refs [k] [l]` donde `l` forma un bucle de 0 a `num_refs [k] - 1`.

// `picture_dependency_maps [N] [N]`, `num_refs [N]`, `refs [N] [N]` son `build_picture_dependency_maps ()` globales

```
{
    int p, q;
    for (p=0; p<N; p++)
        for (q=0; q<N; q++)
            picture_dependency_maps[p][q] = 0;
    for (p=0; p<N; p++)
    {
        for (q=0; q<num_refs[p]; q++)
            add_dep(refs[p][q], p);
    }
}
```

`add_dep(m, n)`

```
{
    int p, q;
    picture_dependency_map[m][n] = 1;
    if (num_refs[m] == 0)
        return;
    for (p=0; p<num_refs[m]; p++)
        add_dep(refs[m][p], n);
}
```

De acuerdo con diversas realizaciones de los presentes principios, proponemos mejoras al conjunto de parámetros de secuencia (SPS) existente en la extensión de codificación de video de múltiples vistas (MVC) del estándar AVC de MPEG-4. El conjunto de parámetros de secuencia mejorado puede indicar la información de orden de codificación de vista y permitir la asignación de un identificador de vista (`view_id`) flexible. Se considera flexible ya que el identificador de vista (`view_id`) es enviado explícitamente en la sintaxis de alto nivel en lugar de tratar la variable de bucle como el identificador de vista. Por supuesto, tal como se indicó anteriormente, aunque los presentes principios se describen principalmente en este documento con respecto a la implementación de mejoras en el conjunto existente de parámetros de secuencia en la extensión de la codificación de video de múltiples vistas

5 del estándar AVC de MPEG-4, dichas mejoras de señalización relacionadas con indicar la información de orden de codificación de vista y que permiten la asignación de un identificador de vista (*view_id*) flexible no se limitan únicamente al conjunto de parámetros de secuencia y pueden ser implementadas mediante una sintaxis de alto nivel al nivel del encabezado del segmento, al nivel del conjunto de parámetros de imagen (PPS), al nivel del conjunto de parámetros de vista (VPS), al nivel del encabezado de la unidad de capa de abstracción de la red (NAL), y en un mensaje de información suplementaria de mejora (SEI), manteniendo al mismo tiempo el espíritu de los presentes principios.

A continuación, se proporcionará una descripción con respecto a una realización de los presentes principios.

10 La TABLA 2 muestra la sintaxis propuesta de la extensión de la codificación de video de múltiples vistas (MVC) del conjunto de parámetros de secuencia de acuerdo con la realización.

TABLA 2

seq_parameter_set_mvc_extension() {	C	Descriptor
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_I0[i]		ue(v)
for(j = 0; j < num_anchor_refs_I0[i]; j++)		
anchor_ref_I0[i][j]		ue(v)
num_anchor_refs_I1[i]		ue(v)
for(j = 0; j < num_anchor_refs_I1[i]; j++)		
anchor_ref_I1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_I0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_I0[i]; j++)		
non_anchor_ref_I0[i][j]		ue(v)
num_non_anchor_refs_I1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_I1[i]; j++)		
non_anchor_ref_I1[i][j]		ue(v)
}		
}		

15 En la realización, proponemos utilizar la variable *i* de bucle para indicar el orden de codificación de la vista. La variable *i* de bucle se incrementa siempre para cada vista. Por lo tanto, el número de orden de codificación de vistas es único para cada vista.

20 Tal como se indicó anteriormente, la técnica anterior indica la variable *i* de bucle como *view_id* y forma un bucle de 0 a *num_view_minus_1*. Por el contrario, utilizamos la variable *i* de bucle como el orden de codificación de vista y formal bucle de 0 a *num_view_minus_1*. Además, señalizamos el *view_id* dentro del bucle. Por lo tanto, podemos asignar cualquier *view_id* a las vistas y no están restringidas a la variable de bucle como es el caso en la extensión actual de codificación de video de múltiples vistas del estándar AVC de MPEG-4. La semántica de las sintaxis se redefine como:

num_views_minus_1 plus 1 identifica el número total de vistas codificadas en el flujo de bits. El valor de *number_of_view_minus_1* estará en el rango de 0 a 1023.

25 **view_id [i]** especifica el *view_id* de la vista con el orden de codificación indicado por *i*.

num_anchor_refs_I0 [i] especifica el número de referencias de predicción entre vistas para list0 para imágenes de vínculo con *view_id* igual a *view_id [i]*. El valor de *num_anchor_refs_I0 [i]* será menor o igual que *num_ref_frames*.

anchor_ref_I0 [i] [j] identifica el view_id de la vista que se utiliza como la referencia de orden j de list0, para las imágenes de vínculo de la vista con view_id igual a view_id [i].

num_anchor_refs_I1 [i] especifica el número de referencias de predicción entre vistas para list1 de la imagen de vínculo con view_id igual a view_id [i]. El valor de num_anchor_refs I1 [i] será menor o igual que num_ref_frames.

5 **anchor_ref_I1** [i] [j] identifica el view_id de la vista que se utiliza como la referencia de orden j de list1, para las imágenes de vínculo de la vista con view_id igual a view_id [i].

num_non_anchor_refs_I0 [i] especifica el número de referencias de predicción entre vistas para list0 para imágenes no de vínculo con view_id igual a view_id [i]. El valor de num_anchor_refs I0 [i] será menor o igual que num_ref_frames.

10 **non_anchor_ref_I0** [i] [j] identifica el view_id de la vista que se utiliza como la referencia de orden j de list0, para las imágenes no de vínculo de la vista con view_id igual a view_id [i].

num_non_anchor_refs_I1 [i] especifica el número de referencias de predicción entre vistas para list1 para imágenes no de vínculo con view_id igual a view_id [i]. El valor de num_anchor_refs I1 [i] será menor o igual que num_ref_frames.

15 **non_anchor_ref_I1** [i] [j] identifica el view_id de la vista que se utiliza como la referencia de orden j de list1, para las imágenes no de vínculo de la vista con view_id igual a view_id [i].

La ventaja de la realización es que los view_id que se asignan a las vistas no tienen ninguna restricción excepto por su rango tal como se define en la semántica. La especificación actual de la extensión de codificación de video de múltiples vistas siempre inicia los view_id desde 0 y los incrementa en 1. Al enviar explícitamente view_id, esta restricción ya no es necesaria en la implementación de acuerdo con los presentes principios.

20

Durante el proceso de decodificación, una imagen puede requerir o no imágenes de referencia entre vistas. Para determinar las imágenes de referencia entre vistas requeridas para una determinada imagen, se llevan a cabo las siguientes etapas de acuerdo con una realización de los presentes principios: (1) leer el view_id de la imagen actual del encabezado de la unidad NAL; (2) buscar este view_id en la matriz view_id [] de SPS (esta matriz tiene el view_id almacenado en el orden de codificación. Por lo tanto, el índice de esta matriz indica el orden de codificación); (3) determinar el índice (i) del view_id de la imagen actual, este es el número de orden de codificación; y (4) utilizar este valor de índice (i) para indexar en otras matrices del conjunto de parámetros de secuencia para determinar las referencias entre vistas necesarias para la imagen actual.

25

Los presentes principios son asimismo útiles para determinar si se ha perdido una imagen durante la transmisión. Esto puede hacerse de la siguiente manera. En la especificación actual de la extensión de codificación de video de múltiples vistas (MVC), sabemos que se realiza la codificación de primero en el tiempo. Antes del inicio de la decodificación para un instante de tiempo particular, podemos configurar un contador view_num en 0. Para cada imagen que se recibe para este instante de tiempo, incrementamos el valor de view_num. Antes de comenzar a decodificar esta imagen, indexamos en la matriz de view_id [view_num] del conjunto de parámetros de secuencia utilizando el contador view_num. Determinamos el view_id correspondiente a este índice. Si este view_id es igual al view_id de la imagen actual, entonces no ha habido pérdida. Sin embargo, si estos view_id son diferentes, entonces sabemos que el view_id indexado se perdió. Esta imagen puede ocultarse entonces mediante un algoritmo apropiado de ocultación de errores. También podemos saber cuántas vistas se han perdido al incrementar view_id hasta que llegar a (coincidir con) el view_id de la imagen actual.

30

40 Para el ejemplo de la figura 3, el view_id se asigna incrementalmente de arriba abajo. El mapeo entre view_num y view_id es el siguiente:

view_num	0	1	2	3	4	5	6	7
view_id	0	2	1	4	3	6	5	7

La construcción de la información del mapa de dependencia a partir de la información actual de dependencia se podría simplificar con el orden de codificación disponible. En el siguiente ejemplo, la indicación de un vínculo / no vínculo se elimina, dado que el algoritmo es aplicable a ambos.

45 Se supone que estamos utilizando estructuras estáticas de datos. Es deseable tener la siguiente información para construir los picture_dependency_maps:

- num_refs [k], indicando el número de referencias entre vistas para la vista con view_id igual a k
- la lista de view_id a los que hace referencia la vista con view_id igual a k: refs [k] [l] donde l forma un bucle de 0 a num_refs [k] - 1.

- view_in_coding_order [k], indicando la lista de view_id en el orden de codificación

// picture_dependency_maps [N] [N], view_in_coding_order [N], num_refs [N], refs [N] [N] son build_picture_dependency_maps_in_order () globales

```
{
    int p, q, pp, s;
    for (p=0; p<N; p++)
        for (q=0; q<N; q++)
            picture_dependency_maps[p][q] = 0;
    for (p=0; p<N; p++)
    {
        pp = view_in_coding_order[p];
        for (q=0; q<num_refs[pp]; q++)
        {
            for (s=0; s<N; s++)
                if (picture_dependency_maps[s][ refs[pp][q] ])
                    picture_dependency_maps[s][pp] = 1;
            picture_dependency_maps[ refs[pp][q] ][pp] = 1;
        }
    }
}
```

5 Volviendo a la figura 4, un método de codificación de contenido de video de múltiples vistas a modo de ejemplo se indica en general mediante el número de referencia 400.

El método 400 incluye un bloque de inicio 405 que pasa el control a un bloque funcional 410. El bloque funcional 410 lee el archivo de configuración del codificador, y pasa el control a un bloque funcional 415. El bloque funcional 415 configura view_direction, view_level y view_id en valores definidos por el usuario, y pasa el control a un bloque funcional 420. El bloque funcional 420 configura de manera flexible view_id [i] (de manera diferencial) en el encabezado del segmento, en el conjunto de parámetros de secuencia (SPS), en el conjunto de parámetros de imagen (PPS), en el conjunto de parámetros de vista (VPS), en la capa de abstracción de la red (NAL), y/o en un mensaje de información suplementaria de mejora (SEI), en base al orden de codificación de la vista (del archivo de configuración del codificador), configura los otros parámetros del conjunto de parámetros de secuencia en base al archivo de configuración del codificador y pasa el control a un bloque funcional 425. El bloque funcional 425 permite que el número de vistas sea igual a una variable N, inicializa una variable i y una variable j a cero, y pasa el control a un bloque de decisión 430. El bloque de decisión 430 determina si i es menor que N. Si es así, entonces el control pasa a un bloque funcional 435. De lo contrario, el control pasa a un bloque funcional 485.

20 El bloque de decisión 435 determina si j es o no menor que el número de imágenes en la vista i. Si es así, entonces el control pasa a un bloque funcional 440. De lo contrario, el control pasa al bloque funcional 485.

25 El bloque funcional 440 comienza a codificar el macrobloque actual (MB) y pasa el control a un bloque funcional 445. El bloque funcional 445 elige el modo del macrobloque y pasa el control a un bloque funcional 450. El bloque funcional 450 codifica el macrobloque actual y pasa el control a un bloque de decisión 455. El bloque de decisión 455 determina si todos los macrobloques han sido codificados o no. Si es así, entonces el control pasa a un bloque funcional 460. De lo contrario, el control se devuelve al bloque funcional 440.

30 El bloque funcional 460 incrementa la variable j y pasa el control al bloque funcional 465. El bloque funcional 465 incrementa frame_num y el recuento de orden de imagen (POC – Picture Order Count, en inglés), y pasa el control a un bloque de decisión 470. El bloque de decisión 470 determina si se señala o no el conjunto de parámetros de secuencia (SPS) y/o el conjunto de parámetros de imagen (PPS) dentro de banda. Si es así, el control pasa a un bloque funcional 475. De lo contrario, el control pasa a un bloque funcional 490.

El bloque funcional 475 envía el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS) y/o el conjunto de parámetros de vista (VPS) a un archivo (es decir, dentro de la banda), y pasa el control a un bloque funcional 480. El bloque funcional 480 escribe el flujo de bits en un archivo o transmite el flujo de bits a través de una red y pasa el control a un bloque final 499.

35 El bloque funcional 485 incrementa la variable i, restablece frame_num y el recuento de orden de imagen (POC) y devuelve el control al bloque de decisión 430.

El bloque funcional 490 envía el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), y/o el conjunto de parámetros de vista (VPS) fuera de banda, y pasa el control al bloque funcional 480.

Volviendo a la figura 5, un método a modo de ejemplo para descodificar el contenido de video de múltiples vistas se indica en general mediante el número de referencia 500.

5 El método 500 incluye un bloque de inicio 505 que pasa el control a un bloque funcional 510. El bloque funcional 510 analiza `view_id`, `view_direction`, y `view_level` del encabezado del segmento, el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el conjunto de parámetros de vista (VPS), el encabezado de la unidad de abstracción de la red (NAL) y/o un mensaje de información de mejora suplementaria (SEI), y pasa el control a un bloque funcional 515. El bloque funcional 515 analiza `view_id [i]` (codificado en modo diferencial) del encabezado del segmento, el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el conjunto de parámetros de vista (VPS), el encabezado de la unidad de abstracción de red (NAL) y/o un mensaje de información de mejora suplementaria (SEI) donde la variable `i` de bucle indica el orden de codificación de la vista y pasa el control a un bloque funcional 520. El bloque funcional 520 analiza otros parámetros del conjunto de parámetros de secuencia, y pasa el control a un bloque funcional 525. El bloque funcional 525 utiliza `view_direction`, al `view_level` y `View_id` para determinar si la imagen actual necesita ser descodificada (verificación de dependencia), y pasa el control a un bloque de decisión 530. El bloque de decisión determina si la imagen actual necesita o no ser descodificada. Si es así, entonces el control pasa a un bloque de decisión 540. De lo contrario, el control pasa a un bloque funcional 535.

El bloque funcional 540 determina si el recuento de orden de imagen (POC) de la imagen actual es igual al recuento de orden de imagen (POC) de la imagen anterior, es decir, $POC(\text{actual}) = POC(\text{anterior})$. Si es así, el control pasa a un bloque funcional 545. De lo contrario, el control pasa a un bloque funcional 550.

20 El bloque funcional 545 configura `view_num` igual a cero y pasa el control al bloque funcional 550. El bloque funcional 550 indexa la información de `view_id` en el nivel alto para determinar el orden de codificación de vista, incrementa el número de vista y pasa el control a un bloque de decisión 555. El bloque de decisión 555 determina si la imagen actual está o no en el orden de codificación esperado. Si es así, el control pasa a un bloque funcional 560. De lo contrario, el control pasa a un bloque funcional 590.

25 El bloque funcional 560 analiza el encabezado del segmento y pasa el control al bloque funcional 555. El bloque funcional 555 analiza el modo del macrobloque (MB), el vector de movimiento (`mv` – Motion Vector, en inglés), y el índice de referencia (`ref_idx`), y pasa el control a un bloque funcional 570. El bloque funcional 570 descodifica el macrobloque actual, y pasa el control a un bloque de decisión 575. El bloque de decisión 575 determina si todos los macrobloques han sido o no descodificados en la imagen actual Si es así, el control pasa a un bloque funcional 580. De lo contrario, el control vuelve al bloque funcional 565.

El bloque funcional 580 inserta la imagen actual en la memoria temporal de imágenes descodificadas (DPB – Decoded Picture Buffer, en inglés) y pasa el control a un bloque de decisión 585. El bloque de decisión 585 determina si todas las imágenes han sido o no descodificadas. Si es así, entonces el control pasa a un bloque final 599. De lo contrario, el control vuelve al bloque funcional 560.

35 El bloque funcional 590 oculta la imagen actual y devuelve el control al bloque funcional 535.

A continuación, se proporcionará una descripción de otra realización de los presentes principios.

40 En la realización, señalizamos el orden de codificación de vista explícitamente en el flujo de bits. La información de orden de codificación de vista puede ser indicada en cualquier sintaxis de alto nivel que incluye, pero no se limita a, sintaxis al nivel del encabezado del segmento, al nivel del conjunto de parámetros de secuencia (SPS), al nivel del conjunto de parámetros de imagen (PPS), al nivel del conjunto de parámetros de vista (VPS), al nivel del encabezado de la unidad de abstracción de red (NAL) y en un mensaje de información suplementaria de mejora (SEI).

45 La TABLA 3 muestra la sintaxis propuesta de la extensión de la codificación de video de múltiples vistas (MVC) del conjunto de parámetros de secuencia de acuerdo con la realización. Por lo tanto, la TABLA 3 ilustra la inserción del orden de codificación de vista para la especificación actual de la extensión de la codificación de video de múltiples vistas (MVC). Esto mantiene la estructura del diseño del conjunto de parámetros de secuencia en la especificación actual de la extensión de la codificación de video de múltiples vistas (MVC), pero añade `view_num` al bucle. Este `view_num` señala el orden de codificación.

TABLA 3

	C	Descriptor
seq_parameter_set_mvc_extension() {		
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_num[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

La

La semántica recién agregada se define de la siguiente manera:

5 **view_num [i]** especifica el orden de codificación de vista de view_id igual a i. view_num se incrementará en uno para cada vista codificada en el orden de codificación.

Para el ejemplo de la figura 3, el view_id se asigna de modo incremental de arriba abajo. El view_num se asigna en función del orden de codificación de la vista como:

view_id	0	1	2	3	4	5	6	7
view_num	0	2	1	4	3	6	5	7

10 Observamos que, en el diseño actual, codificamos una referencia entre vistas para la vista actual utilizando el view_id absoluto en lugar de la diferencia entre el view_id actual y el view_id de referencia. Dado que en la mayoría de los casos la referencia razonable entre vistas debe ser la vista más cercana a la vista actual, podemos codificar la diferencia entre el view_id actual y el view_id de referencia entre vistas.

Aplicación para la detección de errores:

15 Un ejemplo que involucra la utilización del view_num propuesto para detectar una imagen perdida es el siguiente. En la codificación de primero en el tiempo, en el instante de tiempo T8, recibimos un paquete en orden de view_id, 0 1 4 3 5 7. Después de recibir view_id 0 y view_id 1, sabemos que view_id 2 se ha perdido, porque view_id 2 debe ser recibido antes de view_id 1. A continuación, obtenemos view_id 4, 3 y 5, y sabemos que view_id 6 se ha perdido, ya que view_id 6 debe ser recibido antes de view_id 5.

20 Podemos determinar si la imagen que falta en view_id 2 y view_id 6 se ha perdido debido a un error de transmisión o si falta intencionalmente si utilizamos el ejemplo de sintaxis en la TABLA 3. Por lo tanto, si, intencionalmente, deseamos no codificar ciertas vistas, no necesitamos poner los view_id sin codificar en el conjunto de parámetros de secuencia.

Volviendo a la figura 6, otro método a modo de ejemplo para codificar contenido de video de múltiples vistas se indica en general mediante el número de referencia 600.

- El método 600 incluye un bloque de inicio 605 que pasa el control a un bloque funcional 610. El bloque funcional 610 lee el archivo de configuración del codificador y pasa el control a un bloque funcional 615. El bloque funcional 615 configura view_direction, view_level y view_id en valores definidos por el usuario, y pasa el control a un bloque funcional 620. El bloque funcional 620 configura el view_num [i] de manera flexible en el encabezado del segmento, en el parámetro de secuencia conjunto (SPS), en el conjunto de parámetros de imagen (PPS), en el conjunto de parámetros de vista (VPS), en el encabezado de la unidad de abstracción de la red (NAL) y/o en un mensaje de información complementaria de mejora (SEI), en base al orden de codificación de vista para view_id i (del archivo de configuración del codificador), configura los otros parámetros del conjunto de parámetros de secuencia en base al archivo de configuración del codificador, y pasa el control a un bloque funcional 625. El bloque funcional 625 iguala el número de visitas a una variable N, inicializa una variable i y una variable j a cero, y pasa el control a un bloque de decisión 630. El bloque de decisión 630 determina si i es o no menor que N. Si es así, entonces se pasa el control a un bloque funcional 635. De lo contrario, el control pasa a un bloque funcional 685.
- El bloque de decisión 635 determina si j es o no menor que el número de imágenes en la vista i. Si es así, entonces el control pasa a un bloque funcional 640. De lo contrario, el control pasa al bloque funcional 685.
- El bloque funcional 640 comienza a codificar el macrobloque actual (MB), y pasa el control a un bloque funcional 645. El bloque funcional 645 elige el modo del macrobloque y pasa el control a un bloque funcional 650. El bloque funcional 650 codifica el macrobloque actual y pasa el control a un bloque de decisión 655. El bloque de decisión 655 determina si se han codificado o no todos los macrobloques. Si es así, el control pasa a un bloque funcional 660. De lo contrario, el control se devuelve al bloque funcional 640.
- El bloque funcional 660 incrementa la variable j y pasa el control al bloque funcional 665. El bloque funcional 665 incrementa frame_num y el recuento de orden de las imágenes (POC), y pasa el control a un bloque de decisión 670. El bloque de decisión 670 determina si se señala o no el conjunto de parámetros de secuencia (SPS) y/o el conjunto de parámetros de imagen (PPS) en banda. Si es así, el control pasa a un bloque funcional 675. De lo contrario, el control pasa a un bloque funcional 690.
- El bloque funcional 675 envía el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS) y/o el conjunto de parámetros de vista (VPS) a un archivo (es decir, en banda) y pasa el control a un bloque funcional 680. El bloque funcional 680 escribe el flujo de bits en un archivo o transmite el flujo de bits a través de una red o de varias redes y pasa el control a un bloque final 699.
- El bloque funcional 685 incrementa la variable i, restablece frame_num y el recuento de orden de las imágenes (POC) y devuelve el control al bloque de decisión 630.
- El bloque funcional 690 envía el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), y/o el conjunto de parámetros de vista (VPS) fuera de banda, y pasa el control al bloque funcional 680.
- Volviendo a la figura 7, un método a modo de ejemplo para descodificar contenido de video de múltiples vistas se indica en general mediante el número de referencia 700.
- El método 600 incluye un bloque de inicio 705 que pasa el control a un bloque funcional 710. El bloque funcional 710 analiza view_id, view_direction, y view_level del encabezado del segmento, del conjunto de parámetros de secuencia (SPS), del conjunto de parámetros de imagen (PPS), del conjunto de parámetros de vista (VPS), del encabezado de la unidad de abstracción de la red (NAL) y/o de mensaje de información suplementaria de mejora (SEI), y pasa el control a un bloque funcional 715. El bloque funcional 715 analiza el número de vista [i] del encabezado del segmento, el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS), el conjunto de parámetros de vista (VPS), el encabezado de la unidad de abstracción de la red (NAL) y/o un mensaje de información suplementaria de mejora (SEI), donde la variable i de bucle indica el view_id y pasa el control a un bloque funcional 720. El bloque funcional 720 analiza otros parámetros del conjunto de parámetros de secuencia y pasa el control a un bloque funcional 725. El bloque funcional 725 utiliza view_direction, view_level, y view_id para determinar si la imagen actual necesita ser descodificada (verificar la dependencia), y pasa el control a un bloque de decisión 730. El bloque de decisión determina si la imagen actual necesita o no una descodificación. Si es así, el control pasa a un bloque de decisión 740. De lo contrario, el control pasa a un bloque funcional 735.
- El bloque funcional 740 determina si el recuento del orden de imagen (POC) de la imagen actual es igual al recuento del orden de imagen (POC) de la imagen anterior, es decir, POC (actual) = POC (anterior). Si es así, el control pasa a un bloque funcional 745. De lo contrario, el control pasa a un bloque funcional 750.
- El bloque funcional 745 configura view_num igual a cero y pasa el control al bloque funcional 750. El bloque funcional 750 indexa la información view_id en el nivel alto para determinar el orden de codificación de vista, incrementa view_num y pasa el control a un bloque de decisión 755. El bloque de decisión 755 determina si la imagen actual está o no en el orden de codificación esperado. Si es así, el control pasa a un bloque funcional 760. De lo contrario, el control pasa a un bloque funcional 790.

5 El bloque funcional 760 analiza el encabezado del segmento, y pasa el control al bloque funcional 755. El bloque funcional 755 analiza el macrobloque (MB), el vector de movimiento (mv), y el índice de referencia (ref_idx), y pasa el control a un bloque funcional 770. El bloque funcional 770 descodifica el macrobloque actual, y pasa el control a un bloque de decisión 775. El bloque de decisión 775 determina si todos los macrobloques han sido descodificados o no en la imagen actual. Si es así, entonces el control pasa a un bloque funcional 780. De lo contrario, el control vuelve al bloque funcional 765.

10 El bloque funcional 780 inserta la imagen actual en la memoria temporal de imágenes descodificadas (DPB), y pasa el control a un bloque de decisión 785. El bloque de decisión 785 determina si todas las imágenes han sido descodificadas o no. Si es así, entonces el control pasa a un bloque final 799. De lo contrario, el control se devuelve al bloque funcional 760.

El bloque funcional 790 oculta la imagen actual, y devuelve el control al bloque funcional 735.

15 A continuación, se proporcionará una descripción de algunas de las numerosas ventajas / características de la presente invención, algunas de las cuales se han mencionado anteriormente. Por ejemplo, una ventaja / característica es un aparato que tiene un codificador para codificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits resultante, en el que el codificador envía información de codificación de video al menos a la al menos una imagen en una sintaxis de alto nivel.

20 Otra ventaja / característica es que el aparato tiene el codificador tal como el descrito anteriormente, en el que el elemento de sintaxis de alto nivel está incluido en al menos uno de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un conjunto de parámetros de vista, un mensaje de información de mejora suplementaria, un encabezado del segmento y un encabezado de la unidad de capa de abstracción de la red.

25 Otra ventaja / característica es que el aparato tiene el codificador tal como el descrito anteriormente, en el que el contenido de video de múltiples vistas corresponde a al menos dos vistas, incluida la al menos una vista, y, dicho codificador señala la información de orden de codificación de vista en la sintaxis de alto nivel realizando una pluralidad de iteraciones a un bucle a través de una identificación de vista en la sintaxis de alto nivel para cada una de las al menos dos vistas en orden de codificación.

30 Además, otra ventaja / característica es que el aparato tiene el codificador tal como el descrito anteriormente, en el que el contenido de video de múltiples vistas corresponde a al menos dos vistas, incluida la al menos una vista, y dicho codificador señala la información de orden de codificación de vista en la sintaxis de alto nivel realizando una pluralidad de iteraciones a un bucle a través de un orden de codificación de vista en la sintaxis de alto nivel para cada una de las al menos dos vistas.

Además, otra ventaja / característica es que el aparato tiene el codificador tal como el descrito anteriormente, en el que la sintaxis de alto nivel está señalizada en al menos una de en banda y fuera de banda.

35 Asimismo, otra ventaja / característica es que el aparato tiene el codificador descrito anteriormente, en el que dicho codificador codifica en modo diferencial un identificador de vista para una imagen de referencia entre vistas utilizada para codificar la al menos una imagen, cuando dicho codificador codifica la información de dependencia de vista para al menos una imagen.

Adicionalmente, otra ventaja / característica es que el aparato tiene el codificador tal como el descrito anteriormente, en el que dicho codificador asigna de manera flexible un identificador de vista para la al menos una vista en otra sintaxis de alto nivel.

40 Además, otra ventaja / característica es que el aparato tiene el codificador que asigna de manera flexible un identificador de vista para la al menos una vista en otra sintaxis de alto nivel tal como se ha descrito anteriormente, en el que el otro el elemento de sintaxis de alto nivel se incluye en al menos uno de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un conjunto de parámetros de vista, un mensaje de información de mejora suplementaria, un encabezado de segmento y un encabezado de la unidad de la capa de abstracción de la red.

45 Además, otra ventaja / característica es que el aparato tiene el codificador que asigna de manera flexible un identificador de vista para la al menos una vista en otra sintaxis de alto nivel tal como se ha descrito anteriormente, en el que el contenido de video de múltiples vistas corresponde a al menos dos vistas, incluyendo la al menos una vista, y dicho codificador asigna de manera flexible el identificador de vista para cada una de las al menos dos vistas en la otra sintaxis de alto nivel mediante la realización de una pluralidad de iteraciones para realizar un bucle a través del identificador de vista en la otra sintaxis de alto nivel para cada una de las al menos dos vistas en orden de codificación.

55 Además, otra ventaja / característica es que el aparato tiene el codificador que realiza la pluralidad de iteraciones en un bucle a través del identificador de vista en la otra sintaxis de alto nivel para cada una de las al menos dos vistas en orden de codificación tal como se ha descrito anteriormente, en el que la asignación de manera flexible del

identificador de vista para cada una de las al menos dos vistas permite huecos entre identificadores de vista consecutivos.

5 Asimismo, otra ventaja / característica es que el aparato tiene el codificador que asigna de manera flexible un identificador de vista para la al menos una vista en otra sintaxis de alto nivel tal como se ha descrito anteriormente, en el que el contenido de video de múltiples vistas corresponde a al menos dos vistas, incluida la al menos una vista, y dicho codificador asigna de manera flexible el identificador de vista a cada una de las al menos dos vistas en la otra sintaxis de alto nivel realizando una pluralidad de iteraciones en un bucle a través del orden de codificación de vista en la otra sintaxis de alto nivel para cada una de las al menos dos vistas.

10 Adicionalmente, otra ventaja / característica es que el aparato tiene el codificador que asigna de manera flexible un identificador de vista para la al menos una vista en otra sintaxis de alto nivel tal como se ha descrito anteriormente, en donde la otra sintaxis de alto nivel es señalada en al menos una de en banda y fuera de banda.

Además, otra ventaja / característica es que el aparato tiene el codificador tal como se ha descrito anteriormente, en el que el codificador señala de manera flexible la información de dependencia entre vistas en la sintaxis de alto nivel.

15 Estas y otras características y ventajas de los presentes principios pueden ser fácilmente comprobadas por un experto en la técnica relacionada en base a las enseñanzas de este documento. Debe entenderse que las enseñanzas de los presentes principios pueden ser implementadas en diversas formas de hardware, software, firmware, procesadores para propósitos especiales o combinaciones de los mismos.

20 Más preferiblemente, las enseñanzas de los presentes principios se implementan como una combinación de hardware y software. Además, el software puede implementarse como un programa de aplicación incorporado de manera tangible en una unidad de almacenamiento de programas. El programa de la aplicación puede ser cargado y ejecutado por una máquina que comprende cualquier arquitectura adecuada. Preferiblemente, la máquina se implementa en una plataforma de ordenador que tiene hardware tal como una o más unidades de procesamiento central ("CPU" – Central Processing Unit, en inglés), una memoria de acceso aleatorio ("RAM" – Read Only Memory, en inglés) e interfaces de entrada / salida ("I/O" – Input/Output, en inglés). La plataforma de ordenador puede incluir
25 asimismo un sistema operativo y un código de microinstrucción. Los diversos procesos y funciones descritos en este documento pueden formar parte del código de microinstrucción o parte del programa de aplicación, o cualquier combinación de los mismos, que pueden ser ejecutados por una CPU. Además, se pueden conectar otras unidades periféricas a la plataforma informática, como una unidad de almacenamiento de datos adicional y una unidad de impresión.

30 Se debe entender además que, debido a que algunos de los componentes y procedimientos del sistema constituyente representados en los dibujos adjuntos se implementan preferiblemente mediante software, las conexiones reales entre los componentes del sistema o los bloques funcionales del proceso pueden diferir dependiendo de la manera en que se programen los presentes principios. Dadas las enseñanzas de este documento, un experto en la técnica relacionada podrá contemplar estas y otras implementaciones o
35 configuraciones similares de los presentes principios.

REIVINDICACIONES

1. Medio de almacenamiento que contiene datos de señal de video codificados en el mismo, estando los datos de la señal de video codificados mediante la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, y comprendiendo los datos de la señal de video:

5 un flujo de bits que tiene contenido de video de múltiples vistas e información de sintaxis de alto nivel, incluyendo dicha información de sintaxis de alto nivel un conjunto de parámetros de secuencia, incluyendo el conjunto de parámetros de secuencia:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

10 un orden de codificación de vista e identificador de vista indicado por un parámetro, view_id [i], que especifica el identificador de vista de la vista con el orden de codificación de la vista indicado por el índice i, en donde el índice i varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos una.

2. Medio de almacenamiento según la reivindicación 1, en el que el parámetro view_id [i] de los datos de la señal de video señala explícitamente un identificador de vista que indica una vista de dicho contenido de video de múltiples vistas.

15 3. Medio de almacenamiento según cualquiera de las reivindicaciones 1 a 2, en el que el identificador de vista para cada una de una pluralidad de vistas de dicho contenido de video de múltiples vistas se asigna de modo que se permiten espacios entre identificadores de vista consecutivos.

20 4. Medio de almacenamiento según cualquiera de las reivindicaciones 1 a 3, en el que indicar el orden de codificación de vista mediante el índice i permite la utilización de los parámetros ordenados view_id [i] para determinar si una imagen se ha perdido durante la transmisión.

5. Señal de video formateada para incluir información de acuerdo con la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, incluyendo la señal de video:

25 un flujo de bits que tiene contenido de video de múltiples vistas e información de sintaxis de alto nivel, incluyendo dicha información de sintaxis de alto nivel un conjunto de parámetros de secuencia, en el que el conjunto de parámetros de secuencia incluye:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

30 un orden de codificación de vista e identificador de vista indicado por un parámetro, view_id [i], que especifica el identificador de vista de la vista con el orden de codificación de vista indicado por el índice i, en el que el índice i varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos uno.

6. Método para codificar de acuerdo con la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, que comprende:

35 codificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits que tiene información de sintaxis de alto nivel, incluyendo dicha información de sintaxis de alto nivel:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

40 un orden de codificación de vista e identificador de vista indicado por un parámetro, view_id [i], que especifica el identificador de vista de la vista con orden de codificación de vista indicado por el índice i, en el que el índice i varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos uno.

7. Método para decodificar de acuerdo con la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, que comprende:

45 decodificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas de un flujo de bits que tiene información de sintaxis de alto nivel, incluyendo dicha información de sintaxis de alto nivel:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

un orden de codificación de vista e identificador de vista indicado por un parámetro, `view_id [i]`, que especifica el identificador de vista de la vista con orden de codificación de vista indicada por índice `i`, en el que el índice `i` varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos uno.

5 8. Método según la reivindicación 6 o 7, en donde el identificador de vista para cada una de una pluralidad de vistas de dicho contenido de video de múltiples vistas se asigna de tal manera que se permiten espacios entre identificadores de vista consecutivos.

10 9. Método según cualquiera de las reivindicaciones 6 a 8, en el que indicar el orden de codificación de vista mediante el índice `i` permite la utilización de los parámetros ordenados `view_id [i]` para determinar si una imagen se ha perdido durante la transmisión.

10. Medio legible por máquina que tiene almacenadas en él instrucciones ejecutables por máquina que, cuando son ejecutadas, implementan un método de acuerdo con cualquiera de las reivindicaciones 6 a 9.

11. Aparato para codificar de acuerdo con la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, que comprende:

15 un medio para codificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits que tiene información de sintaxis de alto nivel, incluyendo la información de sintaxis de alto nivel:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

20 un orden de codificación de vista e identificador de vista indicado por un parámetro, `view_id [i]`, que especifica el identificador de vista de la vista con orden de codificación de vista indicado por el índice `i`, en el que el índice `i` varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos una.

12. Aparato para descodificar de acuerdo con la extensión de codificación de video de múltiples vistas de un estándar de codificación de video, que comprende:

25 un medio para descodificar al menos una imagen para al menos una vista correspondiente al contenido de video de múltiples vistas en un flujo de bits que tiene información de sintaxis de alto nivel, incluyendo la información de sintaxis de alto nivel:

un parámetro que representa un número total de vistas codificadas en el flujo de bits menos uno, y

30 un orden de codificación de vista e identificador de vista indicado por un parámetro, `view_id [i]`, que especifica el identificador de vista de la vista con orden de codificación de vista indicado por el índice `i`, en el que el índice `i` varía desde 0 hasta el parámetro que representa el número total de vistas codificadas menos una.

35 13. Aparato según la reivindicación 11 o 12, en el que el identificador de vista para cada una de una pluralidad de vistas de dicho contenido de video de múltiples vistas se asigna de modo que se permiten huecos entre identificadores de vista consecutivos.

14. Aparato según cualquiera de las reivindicaciones 11 a 13, en el que indicar el orden de codificación de vista mediante el índice `i` permite la utilización de los parámetros `view_id [i]` ordenados para determinar si una imagen se ha perdido durante la transmisión.

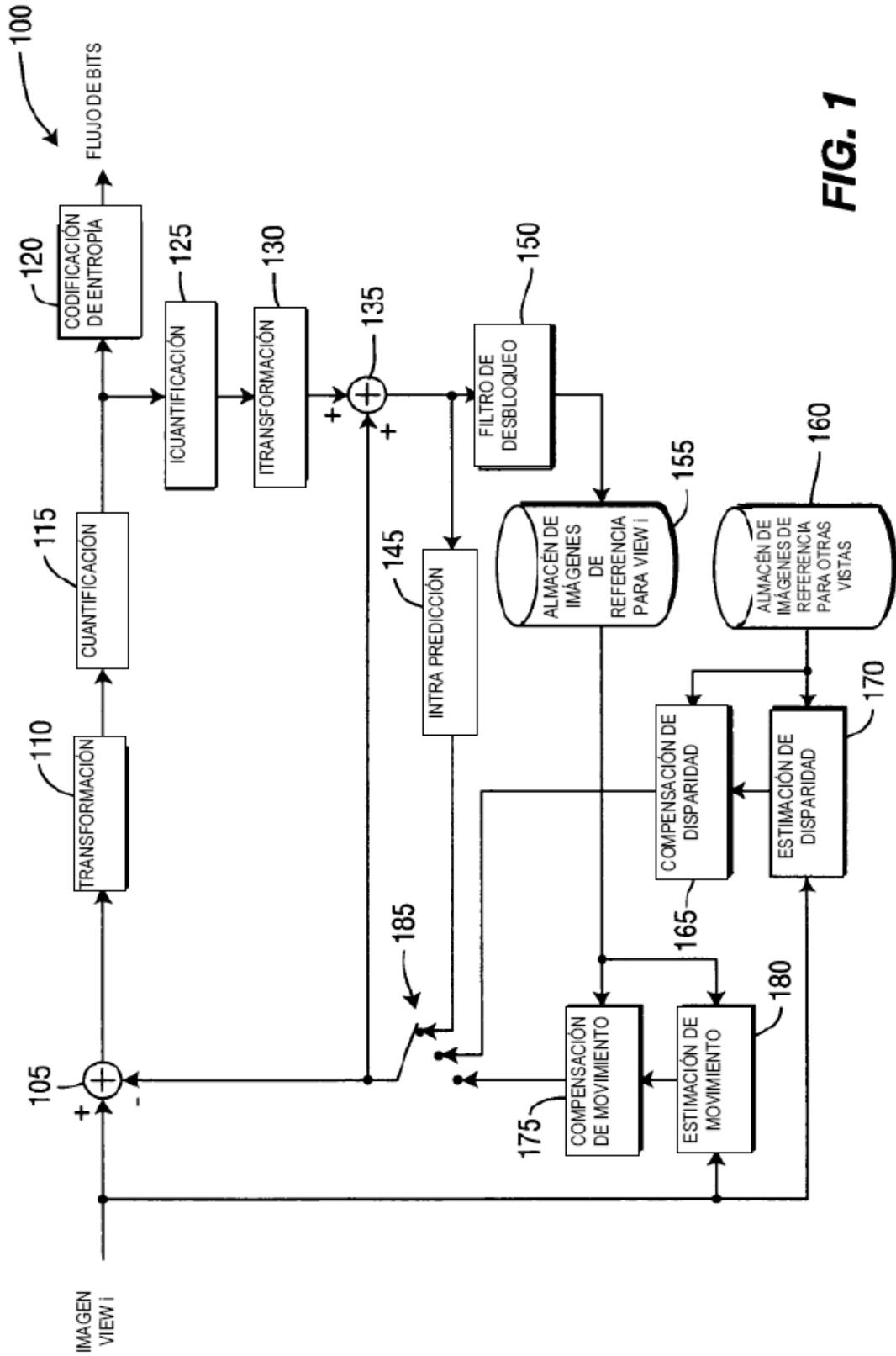


FIG. 1

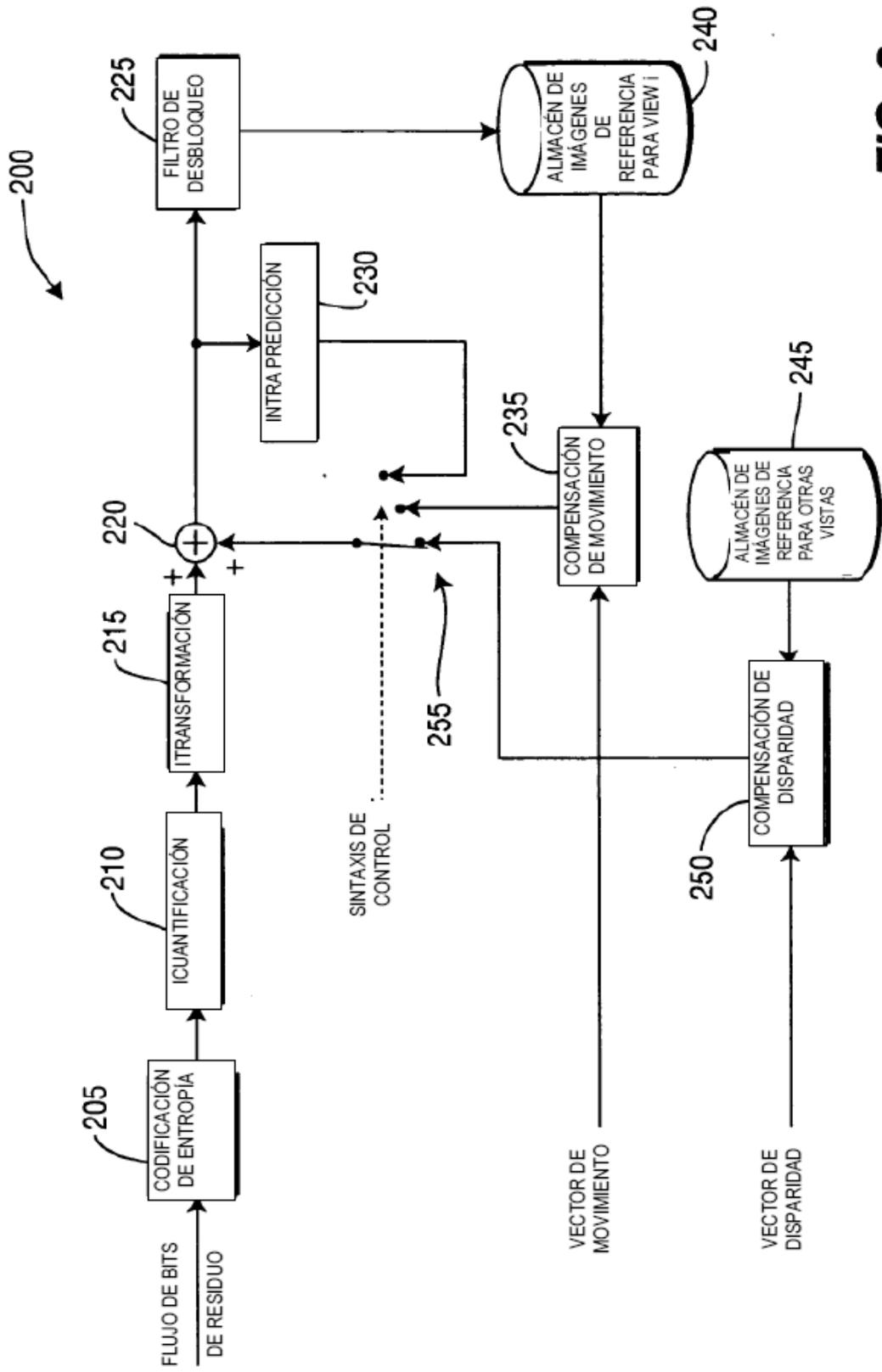


FIG. 2

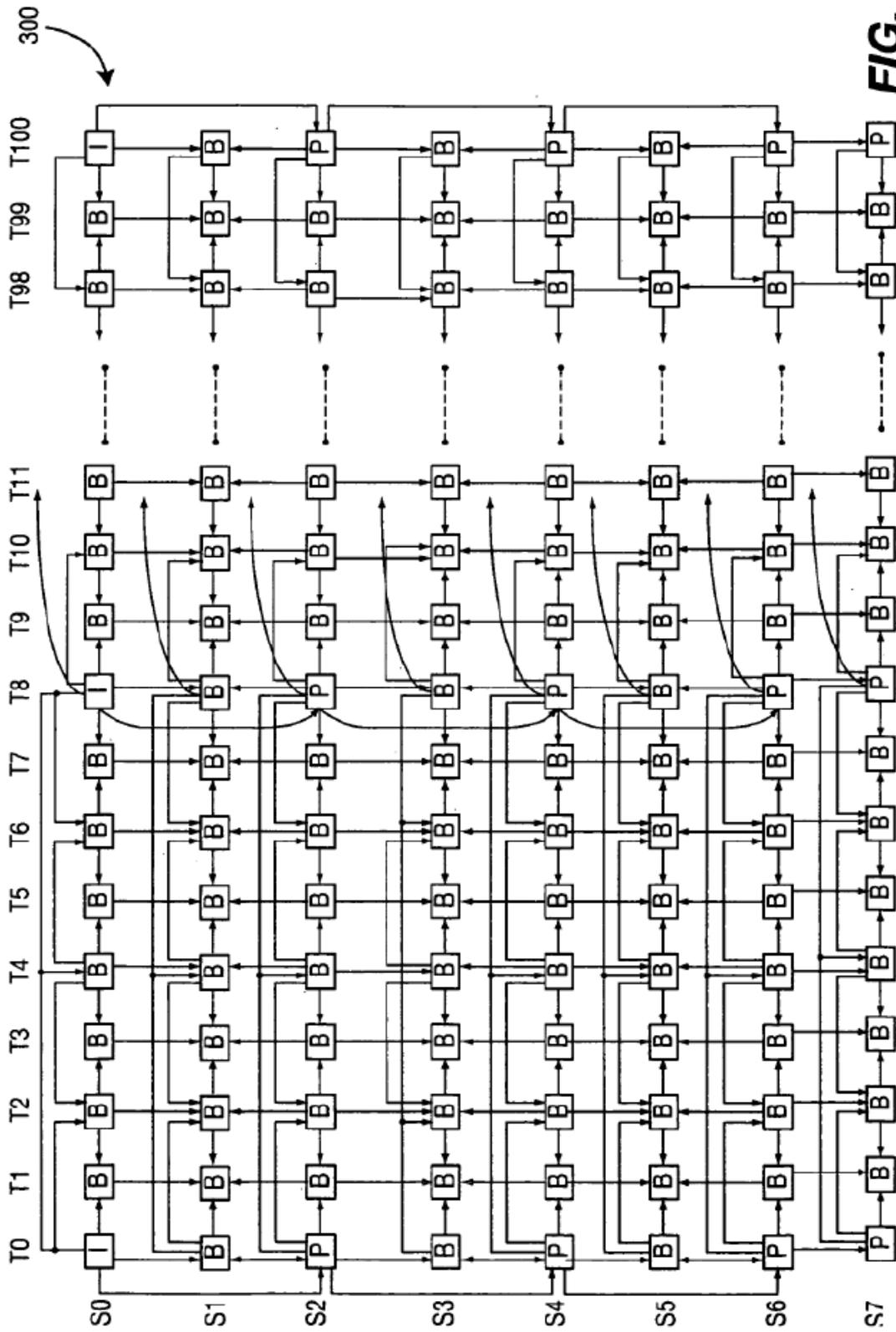


FIG. 3

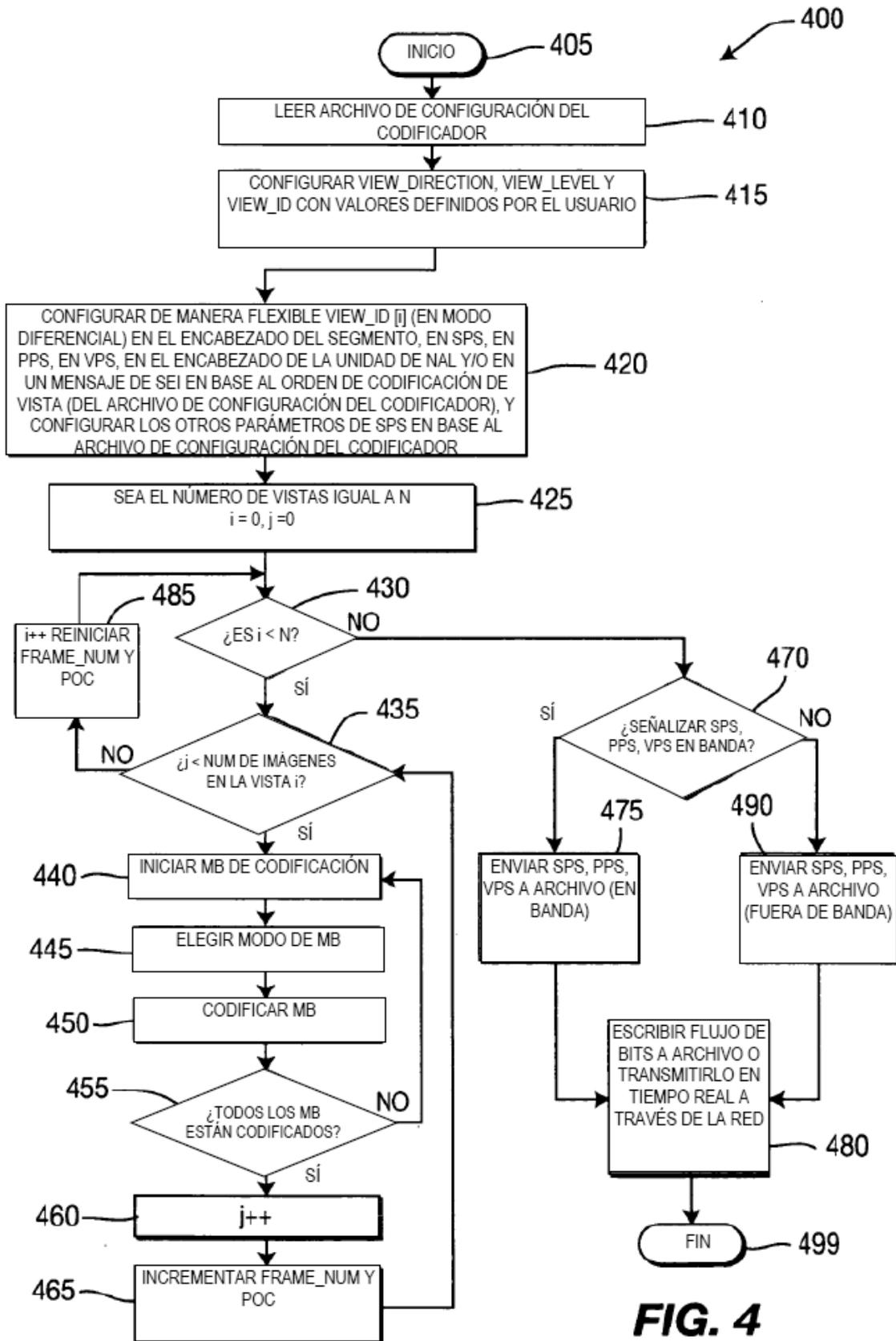


FIG. 4

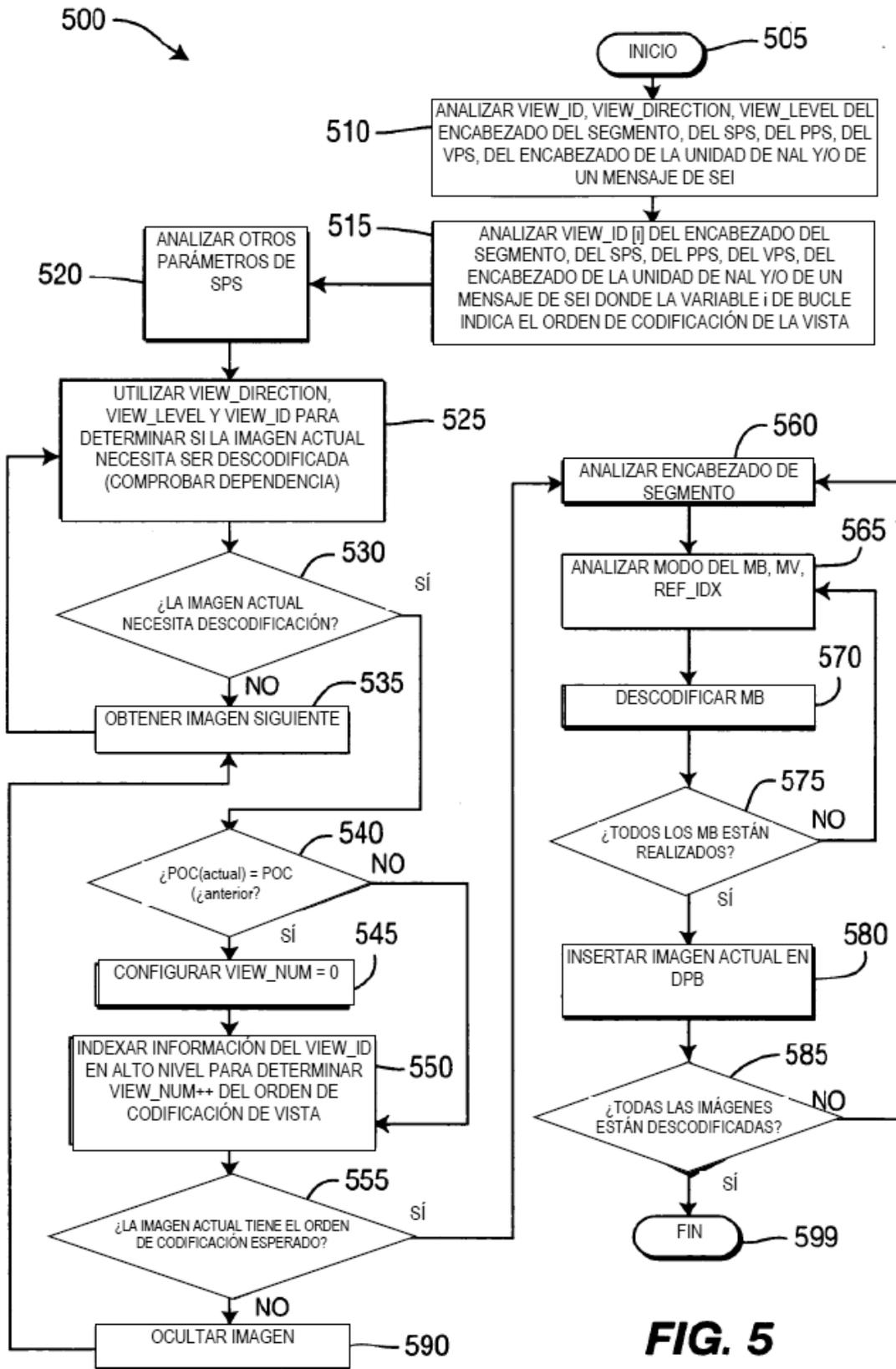


FIG. 5

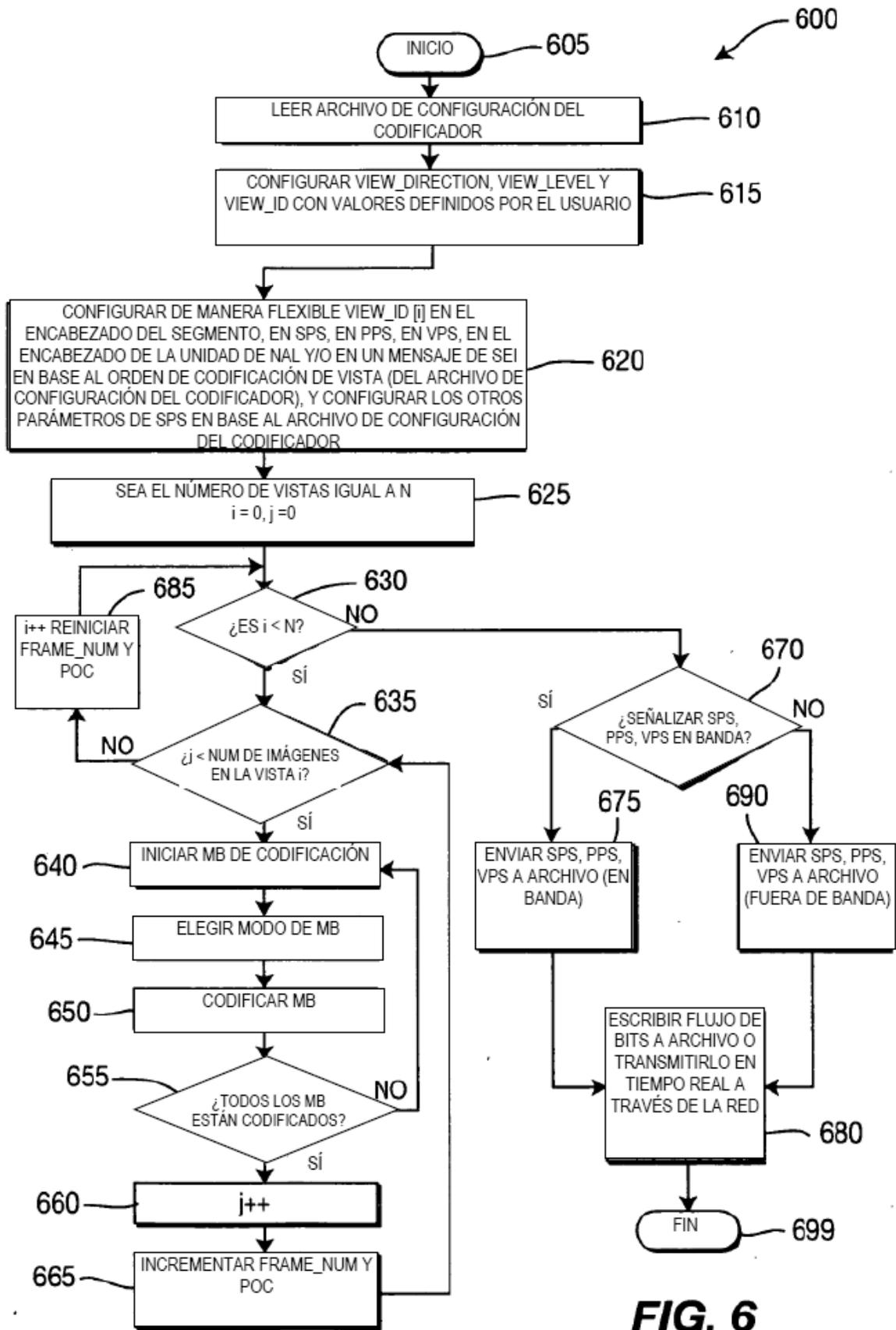


FIG. 6

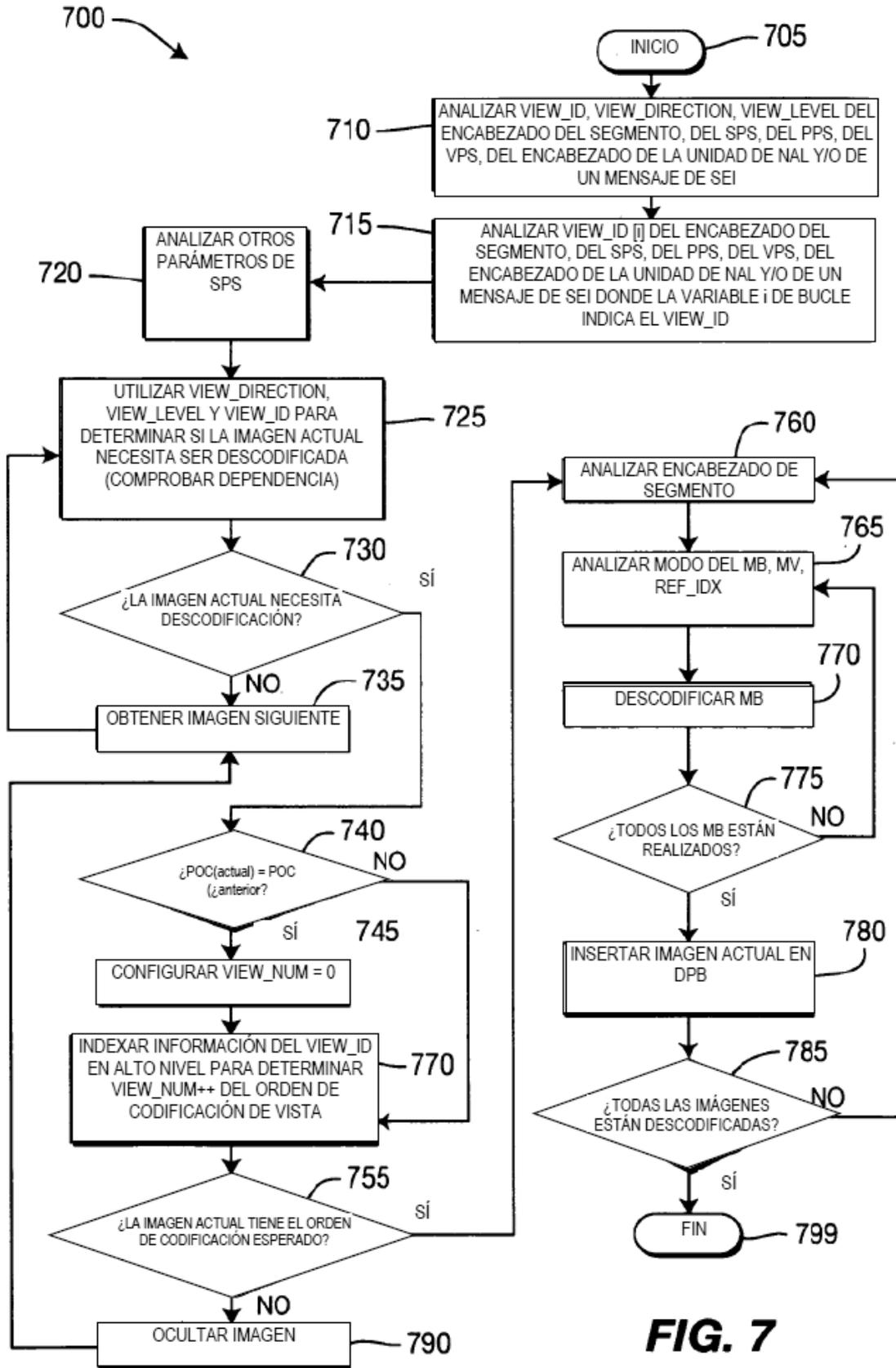


FIG. 7